MAINVIEW® for IMS Offline
Customization and Utilities Guide

Version 3.3.20

November 2002

Restricted Rights Legend

Contacting BMC Software

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

United States and Canada

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

Telephone 713 918 8800 or
800 841 2031

Fax 713 918 8000

Outside United States and Canada

Telephone (01) 713 918 8800

Fax (01) 713 918 8000

Customer Support

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

Support Web Site

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

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

Support by Telephone or E-mail

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

Before Contacting BMC Software

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

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

Contents

Figures

Tables

About This Book

This book describes the procedures required to migrate to version 3.3.20 of MAINVIEW® for IMS Offline, and it documents the features and functions of the Performance Reporter and Transaction Accountant components.

This book is intended for

- IMS system and database administrators who review the activity of each day's IMS session
- system programmers who isolate recurring problems in the IMS environment
- data center managers who make hardware and resource acquisition decisions based on long-term trends

Before you use the Performance Reporter component or the Transaction Accountant component, you must be familiar with the MAINVIEW for IMS Offline environment described in this book.

For information about new features in the current release of MAINVIEW for IMS Offline, see the product release notes, which are available on the BMC Software Support Web pages.

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

How This Book Is Organized

The organization of this book is described in the following table.

Chapter/Appendix	Description
Chapter 1, “Introduction”	describes how the offline components work
Chapter 2, “Getting Started”	describes the information and procedures required to begin using the MVIMS Offline components
Chapter 3, “Migrating to Version 3.3.20 from Version 3.3.10 or 3.3.00”	describes the options you should consider if you are migrating to version 3.3.10 from version 3.3.00
Chapter 4, “Migrating to Version 3.3.20 from Version 3.2”	describes the options you should consider and the changes you need to make if you are migrating to version 3.3.10 from version 3.2
Chapter 5, “Release Compatibility”	describes current product version compatibility with previous releases
Chapter 6, “Implementing MAINVIEW Products in IMS”	explains how to install and implement the modules that are required for MAINVIEW AutoOPERATOR for IMS and MVIMS Offline
Chapter 7, “Event Collector”	describes the component that collects and records the IMS event data used for input to the offline components
Chapter 8, “Customizing with User Exit Routines”	describes the sample user exit routines provided by BMC Software and explains how you can customize the user exit routines to meet your needs
Chapter 9, “Optional Modifications”	describes modifications that can be made to MAINVIEW for IMS
Chapter 10, “Log Edit”	describes the IMFLEDIT program, which processes transaction workload data recorded in the IMS log
Chapter 11, “MVIMS Data and IMS Resource Utilization Files”	describes the IRUF files, which the Performance Reporter and the Transaction Accountant use as input to generate performance and accounting reports
Chapter 12, “IRUF Summarization and Cost Accounting (TASCOSTR)”	explains how the TASCOSTR program is used to summarize the IRUF and produce an IMS resource utilization report
Chapter 13, “IRUF Utilities”	explains how the PRSSELEC utility is used to select records from an input IRUF and how the PRSPRINT utility produces a formatted report describing the contents of the IRUF records
Appendix A, “MVIMS Log Record Layouts”	provides record layout descriptions for transaction and program log records
Appendix B, “IRUF Record Layout Descriptions”	provides IRUF record layout descriptions for the transaction accounting record (TAR), the program accounting record (PAR), and the terminal (LTERM) accounting record (LAR)
Appendix C, “How Product Libraries Should Be Used”	describes how to use distributed and customized parameter, sample, and profile libraries
Appendix D, “Product Initialization Messages”	lists the product initialization messages that have changed with this version of MVIMS Offline

MAINVIEW Product Documentation

MVIMS is integrated with the BMC Software MAINVIEW® architecture. With the MAINVIEW base architecture, authorized users can interrogate any OS/390, CICS, IMS, DB2, or MQSeries subsystem in a sysplex from a single terminal.

This section lists the documents specific to MAINVIEW for IMS products and the documents that provide information common to many MAINVIEW products.

MVIMS Product Library

The MVIMS product library includes the following books.

MAINVIEW for IMS Offline

MAINVIEW for IMS Offline – Customization and Utilities Guide

MAINVIEW for IMS Offline – Performance Reporter Reference Manual

MAINVIEW for IMS Offline – Transaction Accountant Reference Manual

MAINVIEW for IMS Offline – Release Notes

MAINVIEW for IMS Online

MAINVIEW for IMS Online – Customization Guide

MAINVIEW for IMS Online – IPSM Reference Manual

MAINVIEW for IMS Online – Analyzers Reference Manual

MAINVIEW for IMS Online – Monitors and Traces Reference Manual

MAINVIEW for IMS Online – Release Notes

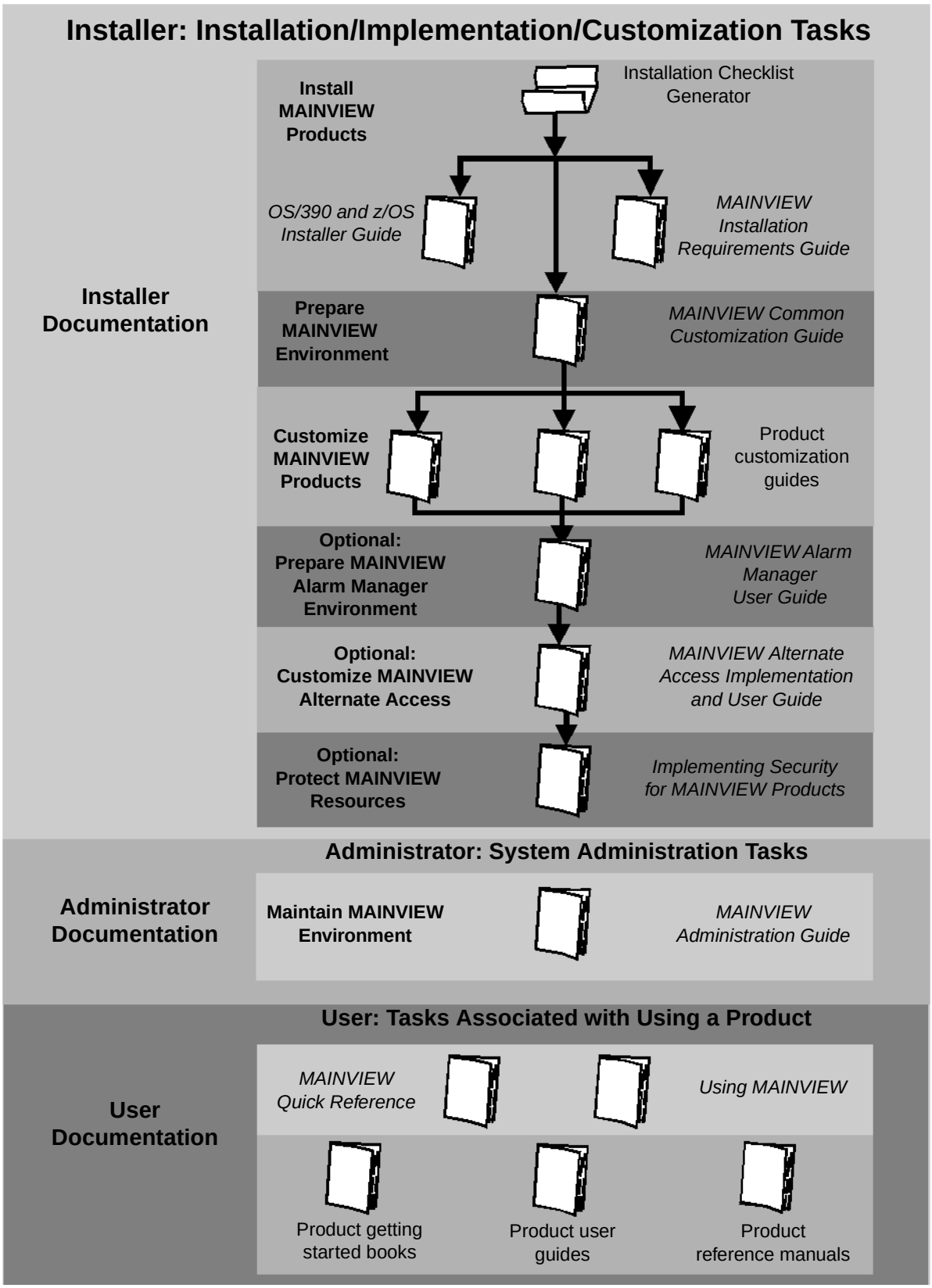
MAINVIEW Product Family Documentation

The books and quick references that provide general information common to many MAINVIEW products are listed and described in the following table.

OS/390 and z/OS Installer Guide	provides information about the installation of BMC Software products on OS/390 and z/OS systems
MAINVIEW Installation Requirements Guide	provides information about installation requirements such as software requirements, storage requirements, and system requirements
MAINVIEW Common Customization Guide	provides instructions for manually customizing the MAINVIEW environment for your products
MAINVIEW Alarm Manager User Guide	explains how to create and install alarm definitions that indicate when exceptions occur in a sysplex
MAINVIEW Alternate Access Implementation and User Guide	explains how to configure, start, and stop VTAM and EXCP AutoLogon sessions to access MAINVIEW products without an active TSO subsystem
Implementing Security for MAINVIEW Products	explains basic MAINVIEW security, enhanced security, and MAINVIEW Alternate Access security
MAINVIEW Administration Guide	provides information about MAINVIEW operations, targets, single-system image contexts, MAINVIEW Alarm Manager, data sets, view customization, and diagnostic facilities
MAINVIEW Quick Reference	introduces the MAINVIEW family of products and lists the commands used to manage the MAINVIEW windows environment
Using MAINVIEW	provides information about working with MAINVIEW products in windows mode and full-screen mode

Note: MAINVIEW messages are documented in the Messages and Codes online display, which you can access by typing MSG in the command line of any MAINVIEW display.

The figure on the next page lists the MAINVIEW product documents and shows how they should be used.



Accessing Product Information

The books that accompany BMC Software products are available in online format and printed format. You can also access product information from product release notes and other product notices.

Online Books

Online books are formatted as Portable Document Format (PDF) files. You can view them, print them, or copy them to your computer using Acrobat Reader 3.0 or later. For information about downloading the free reader from the Web, go to the Adobe Systems site at **http://www.adobe.com**.

You can access online books from the documentation CD that accompanies your product or from the Web. To view online books, visit the support pages of the BMC Software Web site at **http://www.bmc.com/support.html**.

Printed Books

A set of printed books is provided with your product order. To request additional books, go to **http://www.bmc.com/support.html.**

Release Notes and Other Notices

Printed release notes accompany each BMC Software product. Release notes provide current information about new and changed product functions.

A product may also have related technical bulletins that are provided between releases. The latest versions of the release notes and technical bulletins are available on the Web at **http://www.bmc.com/support.html**.

Conventions

The following symbols are used to define command syntax, and they should not be included with a command.

- Brackets [ ] enclose optional parameters or keywords.
- Braces { } enclose a list of parameters, one of which must be chosen.
- A vertical line | separates alternative options, one of which can be chosen.
- An underlined parameter is the default.

The following command syntax conventions also apply:

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

Chapter 1 Introduction

The MAINVIEW for IMS (MVIMS) Online product provides complete IMS monitoring and includes external subsystem monitoring through the IMS Attach Facility.

The MVIMS Offline product has online components that

- measure all IMS-processed transaction, program, and DL/I and DB2 subsystem activity
- collect the IMS event data for offline IMS performance and accounting reporting

How the Offline Components Work

IMS event data is the input to the MVIMS Offline components. The data is recorded in the IMS log as MVIMS transaction and program log records. Data recording is done by an MVIMS component called the Event Collector.

The transaction and program log records contain detailed information about elapsed timings, CPU usage, DL/I database calls and I/O per database, and DB2 subsystem requests per application plan. Most resource usage data is captured at the transaction level and can therefore be attributed to the user.

The MVIMS Offline components (Log Edit, the IRUF utilities, the Performance Reporter, and the Transaction Accountant) process the data collected by the Event Collector, as described in the following sections.

Log Edit

Log Edit extracts the Event Collector records from the IMS log to a file, called the IMS Resource Utilization File (IRUF), for processing by the Performance Reporter and the Transaction Accountant batch report programs.

IRUF Utilities

The PRSSELEC utility selects the IRUF records, and the PRSPRINT utility prints the records.

Performance Reporter

The Performance Reporter

- evaluates and statistically reports IMS workload activity and resources consumed
- provides reporting by message region, program, transaction, database, DB2 plan name, and LTERM
- graphs system and application activity over short or long periods of time for trend analysis reporting
- profiles by month in a calendar format several sets of critical activity and resource usage indicators
- provides detailed analysis of response time, broken down into different elapsed-time components and reported by transaction, program, class, or terminal

Transaction Accountant

The Transaction Accountant provides the detail-level data required for usage-based billing and cost control analysis.

Flow of the Offline Process

This section describes the flow of the MVIMS Offline process.

1. IMS event data is collected by the Event Collector. Because the Event Collector must have access to IMS data, some Event Collector code runs in the IMS regions. The Event Collector writes the collected data to the IMS log as transaction records (X'FA') and program records (X'F9').

2. The Log Edit component is the IMFLEDIT program, which is run against the IMS log tapes to extract, edit, reformat, and write the data to an MVIMS file called the IMS Resource Utilization File (IRUF). IMFLEDIT program execution can produce the following three reports (though the error reports are produced only if associated errors are encountered).

   - Log Edit Audit report
   - Log Edit Error report
   - MVIMS Parameter File Error report

3. The IMS Resource Utilization File (IRUF) created by Log Edit contains transaction, program, and terminal accounting records in detail format, one record per event. You can use the detail IRUF as input to

   - MVIMS Offline components (the Performance Reporter and the Transaction Accountant)
   - the accounting and IRUF summarization component (the TASCOSTR program)
   - MVIMS IRUF utilities (the PRSSELEC and PRSPRINT programs)

4. The TASCOSTR program is the MVIMS IRUF summarization and cost-accounting component. Input to TASCOSTR can be single or multiple detail IRUFs, single or multiple previously summarized IRUFs, or any combination of detail and summarized IRUFs. All input to TASCOSTR must be combined and sorted before processing. Any sort program can be used as a front end to TASCOSTR.

   Different versions of TASCOSTR are distributed with the Performance Reporter and the Transaction Accountant. If both components are installed, the Transaction Accountant version of TASCOSTR is used. The version distributed with the Performance Reporter performs only IRUF summarization. It produces a new summarized IRUF and a Total IMS Resource Usage Analysis report.

The version of TASCOSTR that is distributed with the Transaction Accountant has options that either produce accounting information, perform IRUF summarization, or both. The summarization option produces the same output as the version delivered with the Performance Reporter. The accounting option produces a series of reports, including the Detail Charge-out Analysis, Distribution Charge-out Analysis, and Total IMS Resource Usage Analysis reports.

The summarized IRUF created by TASCOSTR contains one record for all transactions of the same type, submitted by a single customer ID, contained in the input IRUF (or IRUFs).

You can use a summarized IRUF as input to

- MVIMS Offline components (the Performance Reporter and the Transaction Accountant)
- additional TASCOSTR summarization runs
- MVIMS IRUF utilities (the PRSSELEC and PRSPRINT programs)

5. The MVIMS utilities are programs that select and print IRUF data. The select utility is PRSSELEC, and the print utility is PRSPRINT.

   PRSSELEC — The PRSSELEC program is a data selection utility that you can run against either a detail or a summarized IRUF to select specific records. The selected records can be printed in hexadecimal format, written to a file, or both.

   PRSPRINT — When executed, the PRSPRINT program reads and prints every record in the specified input IRUF and creates a formatted report showing the contents of each record in the file.

   Because PRSPRINT has no data selection capabilities, it is usually preceded by a data selection utility such as PRSSELEC.

   The file of selected IRUF records that the PRSSELEC program produces is an IRUF subset. You can use the subset as input to the Performance Reporter and the Transaction Accountant.

6. The Performance Reporter produces reports and plots that you can use to evaluate the performance of IMS and its applications. The reports produced by the Performance Reporter are

   - Total IMS Resource Usage Analysis report
   - Message Region Utilization Analysis reports
   - Program Processing reports
   - Transaction Processing reports
   - General Activity Analysis reports
   - Transaction Response reports
   - Calendar reports

7. The Transaction Accountant produces reports that provide a detailed breakdown of IMS resource usage and usage rates that can be assigned an equivalent monetary value. The Transaction Accountant also produces a summary of IMS activity, grouped by customer, customer/transaction, or transaction.

   The Transaction Accountant can produce reports independently, or it can be interfaced with a site accounting system.

Chapter 2 Getting Started

This chapter describes the information to gather and the procedures to follow before you begin using the MVIMS Offline components.

Read Installation Information

You can find information about installation requirements and procedures in the following books:

- The *OS/390 and z/OS Installer Guide* describes the installation procedures.
- The *MAINVIEW Installation Requirements Guide* lists operating system software requirements, DASD storage requirements, and virtual storage requirements and identifies the function modification IDs (FMIDs) and product target libraries and distribution libraries allocated to MAINVIEW products.
- The *MAINVIEW Common Customization Guide* describes the procedures required to set up the environment for MAINVIEW products.

Specify Event Collector Options

The Event Collector collects IMS event data for the online and offline products. The data collection can be CPU-intensive if all available collections are made. Because site requirements vary, data collection options are built into the Event Collector. With the options, you can limit data collection to information that is useful to your site and worth the cost in CPU overhead.

The options are specified in BBPARM member IMFECP00, and they are documented in Chapter 7, “Event Collector.”

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 MVIMS customer ID. Both the IMS Resource Utilization File (IRUF) and the MVIMS 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: Because transactions from different users can be processed in one program scheduling, the MVIMS 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 on page 8-5) is distributed with MVIMS 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 MVIMS customer ID are listed below. Fields shown in bold type are those that are used in MVIMS reports.

AGN	PERFORMANCE GROUP
ASID	**PLINE**
CLASS	**PROGRAM**
CUSTOMER ID	**PROGRAM TYPE**
FP ROUTE CODE	**PTERM**
MVIMS **SYSID**	**REGION**
IMSID	SMFID
LTERM	**TRANCODE**
MSC DESTINATION ID	**USERID**
MSC ORIGIN ID	VTAM NODE

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 MVIMS summarization, see Chapter 12, “IRUF Summarization and Cost Accounting (TASCOSTR).”
For more information about the use of LTERM in the customer ID, see “Summarization Considerations” on page 12-4.

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 detail IRUFs. (The IMFLEDIT program is run against the IMS log to create IRUFs. For more information about IRUFs, see “Use IRUF Creation and Reporting Strategy” on page 2-4 and Chapter 11, “MVIMS Data and IMS Resource Utilization Files”.)

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.

Use IRUF Creation and Reporting Strategy

BBSAMP member IMFIRUF provides an IRUF creation and reporting strategy and refers to other BBSAMP members that you can use to implement the strategy.

Run Sample Report Set

The JCL required to produce a complete set of sample MVIMS reports is distributed with the MVIMS Offline in BBSAMP member IMFRPTS. Use the IMFRPTS member to produce sample reports. Review the reports to familiarize yourself with the MVIMS output and to verify that the product is working properly.

Note: Before reports can be generated, the Log Edit program IMFLEDIT must be executed to supply the input required by the offline components. For information about executing IMFLEDIT, see page 10-9. For information about executing individual offline programs, see the *MAINVIEW for IMS Offline – Performance Reporter Reference Manual* and the *MAINVIEW for IMS Offline – Transaction Accountant Reference Manual*.

Chapter 3 Migrating to Version 3.3.20 from Version 3.3.10 or 3.3.00

This chapter describes new options and features you should review if you are migrating to MAINVIEW for IMS (MVIMS) Offline version 3.3.20 from version 3.3.00 or 3.3.10.

Migrating from Version 3.3.10

Version 3.3.20 of MVIMS Offline added support for Java region types and ODBA threads. The MVIMS transaction log record (X'FA') and program log record (X'F9') now include the Java region types (Java message processing and Java batch message processing regions) and ODBA threads, which are propagated to the IRUF transaction and program accounting records for inclusion in reports that show region type.

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

Migrating from Version 3.3.00

This section describes product options and features that were introduced with versions 3.3.20 and 3.3.10 of MVIMS Offline.

Java and ODBA Region Support

Version 3.3.20 of MVIMS Offline added support for Java region types and ODBA threads. The MVIMS transaction log record (X'FA') and program log record (X'F9') now include the Java region types (Java message processing and Java batch message processing regions) and ODBA threads, which are propagated to the IRUF transaction and program accounting records for inclusion in reports that show region type.

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

Product Authorization

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

Versions 3.3.10 and 3.3.20 continue to support the product keys method of product activation, and you can continue to use the product keys listed in Table 3-1.

Table 3-1 MVIMS Offline Product Keys

Key	Product or Component
IOF	MAINVIEW for IMS Offline
ITA	MAINVIEW for IMS Transaction Accountant
IPF	MAINVIEW for IMS Performance Reporter

If MVIMS finds one of the product keys listed in Table 3-1 during product initialization and the key is valid, MVIMS will not look for a product authorization table. Therefore, if you want to use a product authorization table to activate MVIMS Offline, you must remove all MVIMS Offline product keys from the BBKEYS member of the BBPARM data set.

If no valid product key is found during initialization, MVIMS looks for the MVIMS Offline product authorization table, IOFTBL3P, in the data set pointed to by the BMCPSWD DD statement in the startup JCL for the IMS control region. If the BMCPSWD DD is not present, MVIMS looks in the STEPLIB, JOBLIB, and LNKLST data sets. If the IOFTBL3P table is found, its authorization information is used to determine whether the CPU is licensed to run MVIMS Offline.

The procedure required to use a product authorization table to activate MVIMS Offline is provided on page 6-9.

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

MSC Clock Synchronization

The Event Collector parameter, MSCCLOCK, was added to BBPARM member IMFECP00 in version 3.3.10 of MVIMS Offline. With IMS 6.1 and later, you can use the MSCCLOCK parameter to specify whether or not MSC systems will, by default, be considered synchronized to the clock of the local IMS system. The MSC systems can be in the same time zone or a different time zone.

If an MSC transaction originates from an MSC system with a clock that is synchronized to the clock of the local IMS, 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').

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.

Additional MSCCLOCK values (*xxxx*,SYNC and *xxxx*,NOSYNC) are provided so that you can specify IDs for individual MSC systems that will be exceptions to the default you use.

For information about how to use the MSCCLOCK parameter, see page 7-9.

UBBPARM Customization Capabilities

With version 3.3.10 and later, you can use the UBBPARM data set to include unique IMFSYS00 and IMFECP00 parameter members for individual IMS systems, eliminating the need to create and allocate an *ibbparm* data set for each IMS system. The IMS-specific settings are defined in members that must have the following name formats:

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

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

For more information about the IMS-specific UBBPARM members, see “Setting Up BBPARM Data Sets” on page 6-3.

Product Initialization Messages

Product initialization messages changed with version 3.3.10. If you have automation that depends on the initialization messages or the sequencing of the messages, please see Appendix D, “Product Initialization Messages.”

Chapter 4 Migrating to Version 3.3.20 from Version 3.2

This chapter describes options you should consider and the changes you need to make if you are migrating to MAINVIEW for IMS (MVIMS) Offline version 3.3.20 from MVIMS version 3.2.

Java and ODBA Region Support

Version 3.3.20 of MVIMS Offline added support for Java region types and ODBA threads. The MVIMS transaction log record (X'FA') and program log record (X'F9') now include the Java region types (Java message processing and Java batch message processing regions) and ODBA threads, which are propagated to the IRUF transaction and program accounting records for inclusion in reports that show region type.

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

Product Authorization

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

Version 3.3.20 continues to support the product keys method of product activation, and you can use the product keys listed in Table 4-1 to activate MVIMS Offline and its components.

Table 4-1 MVIMS Offline Product Keys

Key	Product or Component
IOF	MAINVIEW for IMS Offline
ITA	MAINVIEW for IMS Transaction Accountant
IPF	MAINVIEW for IMS Performance Reporter

If MVIMS finds one of the product keys listed in Table 4-1 during product initialization and the key is valid, MVIMS will not look for a product authorization table. Therefore, if you want to use a product authorization table to activate MVIMS Offline, you must remove all MVIMS Offline product keys from the BBKEYS member of the BBPARM data set.

If no valid product key is found during initialization, MVIMS looks for the MVIMS Offline product authorization table, IOFTBL3P, in the data set pointed to by the BMCPSWD DD statement in the startup JCL for the IMS control region. If the BMCPSWD DD is not present, MVIMS looks in the STEPLIB, JOBLIB, and LNKLST data sets. If the IOFTBL3P table is found, its authorization information is used to determine whether the CPU is licensed to run MVIMS Offline.

The procedure required to use a product authorization table to activate MVIMS Offline is provided on page 6-9.

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

Product Packaging and Keys

The following MVIMS components, which were packaged separately in version 3.2, became part of the MVIMS Offline product with the release of MVIMS Offline version 3.3.00.

IMS Performance Reporter
IMS Transaction Accountant
IMS Performance Reporter Extension for DB2
IMS Transaction Accountant Extension for DB2

If You *Have* Licensed MAINVIEW for IMS Offline

If you have licensed MVIMS Offline and you are using product keys rather than product authorization to run the product, you should use the product key **IOF** to enable all the offline components. The IOF key must be specified in the IMS control region IMFPARM member BBKEYS. The key is not required for report execution.

Note: For information about disabling DB2 data collection, see "Event Collector Option for DB2 Information" on page 4-7.

If You *Have Not* Licensed MAINVIEW for IMS Offline

If you have not licensed MVIMS Offline and you are using product keys rather than product authorization to run the product, you will need to specify the product keys listed in Table 4-2 for the offline components you use. The keys must be specified in the IMS control region IMFPARM member BBKEYS. You do not need to provide keys for report execution.

Table 4-2 Offline Component Product Keys

Key	Component
ITA	MAINVIEW for IMS Transaction Accountant
IPF	MAINVIEW for IMS Performance Reporter
ITD	MAINVIEW for IMS Transaction Accountant Extension for DB2
IPD	MAINVIEW for IMS Performance Reporter Extension for DB2

MSC Clock Synchronization

The Event Collector parameter, MSCCLOCK, was added to BBPARM member IMFECP00 in version 3.3.10 of MVIMS Offline. With IMS 6.1 and later, you can use the MSCCLOCK parameter to specify whether or not MSC systems will, by default, be considered synchronized to the clock of the local IMS system. The MSC systems can be in the same time zone or a different time zone.

If an MSC transaction originates from an MSC system with a clock that is synchronized to the clock of the local IMS, 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').

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.

Additional MSCCLOCK values (*xxxx*,SYNC and *xxxx*,NOSYNC) are provided so that you can specify IDs for individual MSC systems that will be exceptions to the default you use.

For information about how to use the MSCCLOCK parameter, see page 7-9.

Revised IRUF Format

In version 3.3.00 of MVIMS Offline, the maximum number of database trailer (DBT) segments in an IMS Resource Utilization File (IRUF) was increased so that detailed information would be available for up to 225 databases and DD names.

The fixed portion of the transaction accounting record (TAR) is now 676 bytes long.

The maximum number of DBT segments is set with the MAXSEGS parameter in the BBPARM member IMFLEP00. If the MAXSEGS parameter is not specified, its default value of 75 segments is used.

Up to three additional DBT segments (ALLDBS, DB2, and OTHERS) are included in the TAR IRUF DBT if they are found in the DBT of the transaction log record (X'FA').

The TAR DBT section contains DBD (D), DB2 (E), and RESP (R) segments, each 132 bytes long. The PCB (P) segment was moved to the TAR IRUF prefix and is still 192 bytes long.

If you request ddname information (by specifying DBTNAME=DD in the BBPARM member IMFECP00), the DBDNAME DBT segment in the IRUF DBT section will be immediately followed by its associated DDNAME DBT segments.

The only changes required to implement the revised IRUF format are the JCL changes described below.

For more information about the IRUF format, see “CMPFMT32 Format Option” on page 10-8.

JCL Changes

To accommodate the revised IRUF format, the DCB parameters for the RESUTIL DD should be changed to

```
DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974)
```

These DCB parameter changes should also be made to the SORTOUT and DETCOSTS DD statements in the TASCOSTR JCL.

Record Definition Members

The SASIRUF member was updated to reflect the version 3.3.00 (and later) transaction accounting record (TAR) format, and member SASIRF32 was created to describe the version 3.2 format. The COBOL CIMTAR02 member was created to describe the version 3.3.00 (and later) TAR format, and the CIMTAR01 member continues to describe the version 3.2 format.

Version 3.2 Format Option

This section applies to you only if you process IRUF files with your own programs or you use a nonstandard customer ID.

If You Process IRUF Files with Your Own Programs

If you process the IRUF with your own programs and are not yet ready to convert them to support the current record format, you can have the Log Edit program write IRUF records in the version 3.2 format by specifying CMPFMT32=YES in the BBPARM member IMFLEP00.

The option of using the version 3.2 format will be discontinued in the next MVIMS Offline product release.

If You Use a Nonstandard Customer ID

To support more than 50 DBT segments in the IRUF trailer, the LTERM (PCB) segment of the TAR became part of the fixed portion of the record. Relocation of the PCB segment does not affect sites that include LTERM as part of the customer ID.

If you do not use LTERM as part of the customer ID, you will need to continue using the version 3.2 IRUF format, and you must take these steps to do so:

- In the BBPARM member IMFLEP00, specify CMPFMT32=YES.
- If you use Transaction Accountant and you do not include LTERM name as part of the customer ID, you must add a CMPFMT32 parameter to the TASCOSTR EXEC statement. The parameter (,CMPFMT32) is in positions 19 through 27 of the TASCOSTR EXEC statement.

Without the CMPFMT32 parameter in the TASCOSTR EXEC statement, TASCOSTR can process records that are created with the IMFLEP00 CMPFMT32=YES option, but customer IDs that do not include an LTERM name will not be processed properly and the DETCOSTS output file will be in the format used by MVIMS Offline version 3.3.00 and later.

Note: If the CMPFMT32 parameter is included in the TASCOSTR EXEC statement, the IRUF must have been created with the CMPFMT32=YES option in IMFLEP00.

The option of using the version 3.2 format will be discontinued in the next MVIMS Offline product release.

UBBPARM Customization Capabilities

With version 3.3.10 and later, you can use the UBBPARM data set to include unique IMFSYS00 and IMFECP00 parameter members for individual IMS systems, which eliminates the need to create and allocate an *ibbparm* data set for each IMS system. The IMS-specific settings are defined in members that must have the following name formats:

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

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

For more information about the IMS-specific UBBPARM members, see “Setting Up BBPARM Data Sets” on page 6-3.

Event Collector Option for DB2 Information

A FEATURE=NODB2 parameter value was added to PARMLIB member IMFSYS00 in version 3.3.00. You can use the NODB2 parameter value to control the collection of DB2 information.

Set FEATURE=NODB2 if you want to prevent

- tracking of DB2 CPU usage separately from application CPU usage
- creation of a DB2 trailer containing SQL counts

Note: Refer to the PARMLIB member IMFSYSBB for more information.

Product Initialization Messages

Product initialization messages were changed for versions 3.3.20, 3.3.10, and 3.3.00, and new messages were added for version 3.3.00. If you have automation that depends on the initialization messages or the sequencing of the messages, please see Appendix D, “Product Initialization Messages.”

Chapter 5 Release Compatibility

The following sections describe MVIMS Offline version 3.3.20 compatibility with previous releases.

AO Exit

If a previous release of the BMC Software AO exit was installed on the target IMS, make sure that all remnants of the previously installed version are removed by following the instructions in “Deleting Modules from a Prior Release” on page 6-2.

IMS Log Records

Log Edit will process IMS logs created by MVIMS version 3.2 or MVIMS version 3.3.00 and later.

The Performance Reporter and Transaction Accountant components (and associated utilities) will process IRUF records created with either MVIMS version 3.2 or MVIMS version 3.3.00 and later.

Chapter 6 Implementing MAINVIEW Products in IMS

This chapter describes how to install and implement the BMC Software modules that are required for

- MAINVIEW AutoOPERATOR for IMS
- MAINVIEW for IMS Offline

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

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

Note: If you have multiple IMSs, you may want to allocate a BBPARM data set that is unique to an IMS, as described in "Setting Up BBPARM Data Sets" on page 6-3 and "Modifying the IMS Control Region JCL" on page 6-8.

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

Note: All the information in this chapter is included in the *MAINVIEW for IMS Online – Customization Guide* and in the *MAINVIEW for DBCTL Customization Guide*. If you followed the implementation instructions in either of those books, you can disregard this chapter.

Deleting Modules from a Prior Release

Note: You can skip this section if you are installing MVIMS for the first time.

If a previous release of MAINVIEW for IMS, MAINVIEW for DBCTL, or MAINVIEW AutoOPERATOR for IMS was installed on the target IMS, use the information in Table 6-1 and follow the instructions below the table to delete the old modules.

If you copied BBLINK members to a STEPLIB data set using ICOPY, you can use the BBSAMP jobs in Table 6-1 to delete the old modules.

Table 6-1 BBSAMP Jobs to Delete Old Modules

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

To delete the old modules, complete the following steps.

Step 1 From Table 6-1, select the delete job that corresponds to the BMC Software product (or products) and IMS release installed at your site.

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

Step 3 Run the delete job.

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

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

Setting Up BBPARM Data Sets

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

hilevel.*ibbparm*
hilevel.UBBPARM
hilevel.BBPARM

hilevel	The high-level data set name qualifier used at your site.
ibbparm	A user-defined parameter data set that is unique to this IMS You can allocate a separate *ibbparm* data set to contain any members that you want to make unique to the IMS, such as IMFSYS00 and IMFECP00. The *ibbparm* data set must be allocated; it is not created by AutoCustomization. **Note:** If the only members that require customization for an IMS are IMFSYS00, IMFECP00, or both, an *ibbparm* data set is not required. Instead, you can create a renamed (*imsid*SYS) version of the IMFSYS00 member, a renamed (*imsid*ECP) version of the IMFECP00 member, or both, include them in UBBPARM, and customize them to suit your needs (as explained below in UBBPARM).
UBBPARM	A parameter data set that is tailored from the distributed BBPARM data set and is shared by all IMS regions If you used AutoCustomization, you can use the UBBPARM data set created by AutoCustomization. If you did not use AutoCustomization, allocate UBBPARM, copy the distributed BBPARM data set to it, and tailor UBBPARM to suit your needs. In the UBBPARM data set, you can include a copied, renamed version of IMFSYS00 to customize the system parameters for an IMS system, including the SUBSYS parameter, which establishes communication between the IMS and the BBI-SS PAS. The name of an IMS-specific system parameter member must be in the following format: *imsid*SYS (where *imsid* is the IMS ID code) You can also include a copied, renamed version of IMFECP00 to customize the Event Collector parameters for an IMS. The name of an IMS-specific Event Collector parameter member must be in the following format: *imsid*ECP (where *imsid* is the IMS ID code) **Note:** If you do not need to customize *other* UBBPARM members for an IMS system, you do not need to create and allocate an *ibbparm* data set for that system.
BBPARM	The target BBPARM data set distributed by BMC Software

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

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

Customizing BBPARM Members for an IMS System

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

The following procedure customizes the parameter members required to

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

Step 1 Set the system parameters for the IMS system in an IMS-specific UBBPARM member named *imsid*SYS 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. It should be the same subsystem as the one specified in the BBIJNT00 member of the UBBPARM data set or in an *ibbparm* data set.

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

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

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

AOEEXIT and AOIEXIT. Use these parameters to specify the names of user-written AO exit routines. You can also use AOEEXIT and AOIEXIT to specify the order in which AO exit routines get control and whether a return code is padded. (See “Enabling AO Exit Routines” on page 6-10 for more information.)

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

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

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

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

To record BMP and JBP data, specify

BMP=YES (the default)

To time an entire transaction as a single event, specify

CPU=DEPPGM (the default)

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

CICS=YES (the default)

Note: For detailed information about Event Collector parameter selection, see Chapter 7, “Event Collector.”

Step 3 Set up 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 IMS systems. For IMS-specific Log Edit utility parameters, use the IMFLEP00 member in an *ibbparm* data set.

The IMSLEVEL parameter in IMFLEP00, used by the Log Edit utility IMFLEDIT, is required only for MAINVIEW for IMS Offline.

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

IMSLEVEL=5100 | 6100 | 7100 | 8100 | 0000

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

Note: For more information about the Log Edit parameters and their functions, see Chapter 10.

Activating MAINVIEW for IMS Offline

You can use product keys or a product authorization table to activate MVIMS Offline.

Product Keys

If you have licensed MVIMS Offline and you want to use a product key to activate the product, specify product key IOF in BBPARM member BBKEYS.

If you have not licensed MVIMS Offline and you want to use product keys to activate the offline components, you can use the following product keys:

ITA	IMS Transaction Accountant
IPF	IMS Performance Reporter
ITD	IMS Transaction Accountant Extension for DB2
IPD	IMS Performance Reporter Extension for DB2

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

Product Authorization Table

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

Step 1 Create the product authorization table IOFTBL3P, either through AutoCustomization or with a batch job. Both methods are described in the *OS/390 and z/OS Installer Guide*.

Step 2 Add access to the product authorization table (as described on page page 6-9).

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

Modifying the IMS Control Region JCL

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

Allocating a BBPARM Data Set

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

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

Giving IMS Access to the Event Collector

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

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

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

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

ICOPY8 for IMS 5.1

ICOPY9 for IMS 6.1

ICOPY1 for IMS 7.1

ICOPY2 for IMS 8.1

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

Adding Access to a Product Authorization Table

If you are using the IOFTBL3P product authorization table to activate MVIMS Offline, the IMS control regions must be able to access the table.

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

- Concatenate a data set containing the IOFTBL3P product authorization table to the STEPLIB data set in the IMS control region JCL.
- Add the following statement to the IMS control region JCL:

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

- Copy the IOFTBL3P table into RESLIB or BBLINK.
- Add the IOFTBL3P data set to LNKLST.

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

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

Enabling AO Exit Routines

Note: You can skip this section if you have a prior version of MVIMS Offline installed.

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

BMC Software supplies two AO exit routines to do message processing:

- type-1 AO exit routine, DFSAOUE0
- type-2 AO exit routine, DFSAOE00

Note: The IBM *IMS Operations Guide* explains the differences between the initialization and message processing routines.

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

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

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

If you require a DFSAOE00 or DFSAOUE0 exit routine in addition to the exit provided by BMC Software, perform Step 1 and Step 2 on page 6-11.

Step 1 If you have

One DFSAOE00 exit:

Rename it DFSAOE01, or choose a different name and use the AOEEXIT control statements described for multiple DFSAOE00 exits (below).

Multiple DFSAOE00 exits in addition to the BMC Software–supplied exit, or one exit not named DFSAOE01:

You must add one or both of the following AOEEXIT control statements to an *imsid*SYS member in the UBBPARM data set or to an IMFSYS00 member in an *ibbparm* data set.

For example:

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

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

Step 2 If you have

One DFSAOUE0 exit:

Rename it DFSAOUE1, or choose a different name and use the AOIEXIT control statements described for multiple DFSAOE0 exits (below).

Multiple DFSAOUE0 exits in addition to the BMC Software–supplied exit, or one or more exits not named DFSAOUE1:

You must add one or both of the following AOIEXIT control statements to an *imsid*SYS member in the UBBPARM data set or to an IMFSYS00 member in an *ibbparm* data set.

For example:

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

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

Flow of Control

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

- the order in which exits are to be processed
- the exit return code

For example:

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

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

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

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

Enhancement to the IMS DFSAOE00 Interface

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

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

MAINVIEW AutoOPERATOR for IMS Considerations

When you install your user exit with a BMC Software exit, you should keep the following considerations in mind.

- If the BMC Software exit is executed before the user exit and the user exit's return code cancels the processing of additional message segments of a multisegment message, the message segments are also canceled for MAINVIEW AutoOPERATOR. The result is that incomplete IMS messages are passed to the MAINVIEW AutoOPERATOR and LAST SEG LOST messages from BBI. The MAINVIEW AutoOPERATOR AO exit holds the first segment for a certain length of time while waiting for additional segments, which can cause the messages to be processed out of timestamp sequence by MAINVIEW AutoOPERATOR.

- When the user exit is executed before the BMC Software exit, changes to the messages made by the user exit are received by the MAINVIEW AutoOPERATOR AO exit.

  – If the user exit sets the length code of a message (or a segment of a multisegment message) to zero, MAINVIEW AutoOPERATOR does not process the message (or segment).

  – If the user exit sets the length code of the first segment to zero and the return code to 4, all further segments are canceled for MAINVIEW AutoOPERATOR also.

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

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

Chapter 7 Event Collector

he Event Collector is an MVIMS component that collects and records IMS event data. The collected data is used as input to the MVIMS Online and MVIMS Offline components.

The data collected for the Offline components is recorded in the IMS log. The IMS log is processed by Log Edit, which creates the IMS Resource Utilization File (IRUF), the primary file used by the Offline components.

Data Collection Parameters

Data Collection parameters for the Event Collector are described in Table 7-1 on page 7-2, and recovery option parameters are described in Table 7-2 on page 7-10. The parameters specify the data to be collected, the functions to be performed during error recovery, and the diagnostics to be done by MVIMS. Because some of the available collection, recovery, and diagnostic parameter settings can be CPU-intensive, you should use careful consideration when you select them. The parameters, which are specified in BBPARM member IMFECP00, are read and processed by MVIMS at IMS initialization and remain in effect throughout the IMS session.

The default parameter values in Table 7-1 and Table 7-2 are underlined.

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 plus MAINVIEW for IMS usage is 220, as shown in Example 1 on page 7-2.

Example 1:

IMS = 200
Event Collector = 10% of IMS = 200 × 0.10 = 20
Total = IMS + Event Collector = 200 + 20 = 220

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

Example 2:

IMS = 200
Event Collector = 10% of IMS = 200 × 0.10 = 20
Total = IMS + Event Collector + BFALTERS = 200 + 20 + 8 = 228

Table 7-1 lists the Event Collector collection options alphabetically by parameter name.

Table 7-1 Event Collector Data Collection Options (Part 1 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
BHTO	OFF	**Function:** Includes buffer handler time (BHT) data in DL/I CPU time data. **Note:** BHTO=OFF is forced if DBIO option is not BFALTERS.
		Data: BHT data is included in DL/I CPU time data. **Note:** DEDB and MSDB BHT for IMS Fast Path is always included in DL/I CPU time data. 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.
		CPU Usage: Most efficient option.
	ON	**Function:** Collects BHT data separately.
		Data: BHT data is available for the Performance Reporter and the Transaction Accountant.
		CPU Usage: Time that Event Collector adds to IMS usage can be increased 20 to 40 percent; in other words, if total MVIMS overhead is 10 percent of IMS CPU, the option could cause the total overhead to be 12 to 14 percent. This increase depends on the percentage of DL/I database calls in the total IMS workload.

Table 7-1 Event Collector Data Collection Options (Part 2 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
BILLOVHD	NO	**Function:** Includes prior transaction termination, current transaction scheduling, program load, and schedule-to-first DL/I as dependent region overhead CPU time.
		Data: The Performance Reporter, the Transaction Accountant, and the Workload Analyzer trace services report actual program CPU time, resulting in less chargeable CPU time in the Transaction Accountant.
		CPU Usage: Distribution change only.
	YES	**Function:** Includes prior transaction termination, current transaction scheduling, program load, and schedule-to-first DL/I as dependent region chargeable CPU time.
		Data: The Performance Reporter and the Transaction Accountant report increased chargeable CPU time in the Transaction Accountant. **Notes:** • This option may 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.
		CPU Usage: Distribution change only.
	SCHEDDLI	**Function:** Includes prior transaction termination and current transaction scheduling as dependent region overhead CPU time. Includes program load and schedule-to-first DL/I as dependent region chargeable CPU time.
		Data: The Performance Reporter and the Transaction Accountant report increased chargeable CPU time in the Transaction Accountant if BILLOVHD=NO (the default) is used and is changed to BILLOVHD=SCHEDDLI. **Notes:** • This option may 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.
		CPU Usage: Distribution change only.

Table 7-1 Event Collector Data Collection Options (Part 3 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
BMP	YES	**Function:** Collects BMP and JBP transaction and program activity data.
		Data: Data is available for the Performance Reporter, the Transaction Accountant, and workload trace.
		CPU Usage: Event Collector CPU usage depends on the number and activity of all BMPs and JBPs, but usage is higher than if NO or NOCPU is specified.
	NO	**Function:** Does not measure BMP and JBP activity.
		Data: No data is available for the Performance Reporter, the Transaction Accountant, and workload trace.
		CPU Usage: Event Collector CPU usage depends on the number and activity of all BMPs and JBPs, but usage is lower than if YES or NOCPU is specified.
	NOCPU	**Function:** Collects BMP and JBP transaction and program activity data, but not CPU time usage.
		Data: No CPU data is available for BMPs and JBPs. All other data, such as DL/I counts and database accesses, are still available for the Performance Reporter, the Transaction Accountant, and workload trace.
		CPU Usage: CPU time measurement, the largest overhead item, is not taken. This option enables the collection of other statistics.
CICS	YES	**Function:** Combines online and offline functions.
		Data: Data is available for offline batch reports, the Workload Analyzer wait and trace services, and the Workload Monitor services.
		CPU Usage: Usage depends on the number and activity of CICS and ODBA transactions and programs.
	ONLINE	**Function:** Records data for CICS and ODBA threads and sends it to the BBI-SS PAS for processing by the Workload Analyzer wait and trace services and the Workload Monitor services, but does not write the data to the IMS log.
		Data: Data is available for the Workload Monitor services and the Workload Analyzer wait and trace services.
		CPU Usage: Usage depends on the number and activity of CICS and ODBA transactions and programs.
	OFFLINE	**Function:** Records data for CICS and ODBA threads and writes it to the IMS log for batch report processing.
		Data: Data is available for Offline batch components, such as the Performance Reporter and the Transaction Accountant, but not for the Workload Analyzer wait and trace services and the Workload Monitor services.
		CPU Usage: Usage depends on the number and activity of CICS and ODBA transactions and programs.
	NO	**Function:** Does not measure CICS and ODBA thread activity data.
		Data: No data is available for the Performance Reporter and the Transaction Accountant, the Workload Analyzer wait and trace services, and the Workload Monitor services.
		CPU Usage: Usage depends on the number and activity of CICS and ODBA transactions and programs.

Table 7-1 Event Collector Data Collection Options (Part 4 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
CPICDB2	TERM	**Function:** For CPI-C (explicit APPC) conversations, writes a single transaction record for each conversation. **Note:** The CPICDB2 options are in effect for a conversation until the application does an APSB call. When an APSB call is issued, the CPICDLI options are then in effect for the remainder of the application.
		Data: The Workload Analyzer, the Workload Monitor, the Performance Reporter, and the Transaction Accountant report one transaction for each CPI-C conversation.
		CPU Usage: No additional CPU.
	SYNC	**Function:** For CPI-C (explicit APPC) conversations, writes a transaction record for each SRRCMIT.
		Data: The Workload Analyzer, the Workload Monitor, the Performance Reporter, and the Transaction Accountant report one transaction for each CPI-C conversation.
		CPU Usage: Minimal increase occurs at sync point.
CPICDLI	APSB	**Function:** For CPI-C (explicit APPC) conversations, writes a transaction record each time a new PSB is allocated by an APSB call. **Note:** The CPICDLI options override the CPICDB2 options if and when the application does an APSB call.
		Data: The Workload Analyzer, the Workload Monitor, the Performance Reporter, and the Transaction Accountant report one transaction for each APSB call.
		CPU Usage: No additional CPU.
	SYNC	**Function:** For CPI-C (explicit APPC) conversations, writes a transaction record for each SRRCMIT.
		Data: The Workload Analyzer, the Workload Monitor, the Performance Reporter, and the Transaction Accountant report one transaction for each sync point.
		CPU Usage: Minimal increase occurs at sync point.

Table 7-1 Event Collector Data Collection Options (Part 5 of 9)

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

Table 7-1 Event Collector Data Collection Options (Part 6 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
CPUOVHD	YES	**Function:** Collects control region overhead and DLISAS region overhead in the MVIMS program log record (X'F9'), even if CPU=ALL is not specified. **Note:** Control region overhead and DLISAS region overhead are always collected when CPU=ALL is specified; therefore, CPUOVHD=YES has no effect when CPU=ALL is specified. CPUOVHD=YES has no effect on dependent region overhead.
		Data: The overhead CPU fields in the records include control region overhead and DLISAS region overhead values.
		CPU Usage: CPUOVHD does not affect CPU usage.
	REFCPU	**Function:** Collects overhead CPU values in the MVIMS program log record (X'F9') based on the CPU keyword specification.
		Data: REFCPU means REFer CPU. The data collected is determined by the use of the CPU and CPUOVHD parameters together. If you do not want to collect CPU overhead data, use these options: CPUOVHD=REFCPU CPU=NONE If you want to collect all CPU overhead data from control, DLISAS, and dependent regions, use these options: CPUOVHD=REFCPU CPU=ALL If you want to collect CPU overhead data from the dependent region, without collecting DLISAS or control region CPU overhead data, use these options: CPUOVHD=REFCPU CPU=DEP The CPU overhead values are set to zero when no CPU timing is done. In other words, control region overhead CPU is set to zero when CPU=NONE or CPU=DEP. Message region overhead CPU is also set to zero when CPU=NONE.
		CPU Usage: CPUOVHD does not affect CPU usage.
Note: The CPUOVHD parameter applies only to CPU overhead; it does not affect chargeable CPU time.		
DBFP	NO	**Function:** Collects counts of NONKEY WRITES and NO I/O ALTERS and reports them in the database trailer.
		Data: Counts of NONKEY WRITES and NO I/O ALTERS are collected and are available for performance analysis and billing.
	YES	**Function:** Bypasses collection of NONKEY WRITES and NO I/O ALTERS.
		Data: NONKEY WRITES and NO I/O ALTERS counts are not reported in the database trailer. This value is recommended if the statistics are not required for performance analysis or billing.
		CPU Usage: Reduces CPU usage for Fast Path transactions.

Table 7-1 Event Collector Data Collection Options (Part 7 of 9)

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

Table 7-1 Event Collector Data Collection Options (Part 8 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
DBTS4BMP	30	**Function:** Sets maximum number of database trailers allowed per BMP or JBP region. Valid values are 2 to 500 (30 is recommended). If a value is not specified, the value specified for DBTS is the default.
		Data: If a transaction accesses more than this number of databases, resource data is collected in an overflow trailer named OTHERS.
		CPU Usage: None. Affects ECSA requirement for each region.
MSCCLOCK (IMS 6.1 or later)	DEFAULT, NOTSYNC DEFAULT, SYNC	**Function:** Specifies whether or not MSC systems are to be considered, by default, synchronized to the clock of the local IMS. Exceptions to the DEFAULT can be specified with MSCCLOCK=*xxxx*,NOTSYNC or MSCCLOCK=*xxxx*,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. Only one MSCCLOCK=DEFAULT,NOTSYNC | DEFAULT,SYNC record is accepted (the first one specified); if another DEFAULT record is encountered, it will be ignored.
		Data: None.
		CPU Usage: None.
	xxxx, NOTSYNC *xxxx*, SYNC	**Function:** Used to specify exceptions to the MSCCLOCK DEFAULT setting. The value *xxxx* represents an MSC ID, which can range from 0001 to 2036. An MSC exception specification is valid whether it precedes or follows the MSCCLOCK=DEFAULT,NOTSYNC | SYNC record.
		Data: None.
		CPU Usage: None.
SYSID	1	**Function:** Identifies MVIMS system for the Performance Reporter and the Transaction Accountant. Valid values are 1 through 9 and A through Z. Used to identify and select data from other IMS systems.
		Data: None.
		CPU Usage: None.
TELON	NO	**Function:** Disables specific data collection for the TELON application package.
		Data: None.
		CPU Usage: None.
	YES	**Function:** 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 9-3.)
		Data: None.
		CPU Usage: None.

Table 7-1 Event Collector Data Collection Options (Part 9 of 9)

Parameter	Value	Function, Data Collected, and CPU Usage
TRNSYNC	NO	**Function:** Does not write a transaction record at BMP or JBP checkpoint.
		Data: A record is written only per each successful MESSAGE GET UNIQUE or at program end.
		CPU Usage: None.
	YES	**Function:** Writes a transaction record at BMP or JBP checkpoint.
		Data: A record is written.
		CPU Usage: Minimal.

Table 7-2 lists Event Collector recovery options.

Table 7-2 Event Collector Recovery Options

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

Data Collection for Report Program Processing

Use the following parameters to specify the data you want collected in the transaction and program records. Log Edit writes these records to the IMS log in order to create the IRUF file, which is used for batch report processing.

BMP	DBIO
BHTO	BILLOVHD
CPU	DBTS
DBFP	DBTS4BMP

CPU Usage

Parameters that can affect MVIMS CPU usage include

BMP	DBFP
BHTO	DBIO
CPU	DEPREC

The next section describes several combinations of these parameter definitions as examples of standard, minimum, and maximum data collection. The impact of MVIMS data collection on CPU usage is described for each option set.

Parameter Option Sets

This section describes three sets of parameter definitions.

Standard Option Set

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

```
BMP=YES          DBFP=NO
BHTO=OFF         DBIO=IOWAITS
CICS=YES         DEPREC=YES
CPU=DEPPGM
```

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

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

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

```
BMP=YES          DBFP=NO
BHTO=OFF         DBIO=BFALTERS
CICS=YES         DEPREC=YES
CPU=ALL
```

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

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

For the DBIO parameter, the level of data collection provided by the BFALTERS option is rarely used on a daily basis. You can reduce collection overhead by specifying the IOWAITS option instead. You can then specify the BFALTERS option only when you need more data 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 MVIMS CPU usage of 15 to 25 percent of total IMS CPU, though some individual BMPs or JBPs with a large amount of DL/I activity can be higher.

Minimum Option Set

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

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

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 MVIMS 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 MVIMS CPU usage of 3 to 10 percent of total IMS CPU.

IMS and MVIMS CPU Usage Workload Dependencies

When analyzing or estimating IMS and MVIMS CPU usage, you must consider the following criteria.

- Changes in IMS configuration can have a major effect on total CPU usage or on its distribution. For example, changing the IMS LSO option can move a large percentage of CPU from one IMS region to another (dependent, control, DLISAS). In addition, LSO=Y is usually more expensive to run than the other options. Moving from one machine model to another affects CPU usage per transaction. Therefore, comparisons cannot easily be made when such changes have occurred.

- The amount of IMS CPU consumed in areas that are not measured by the Event Collector can have a major impact on the percentage of total CPU usage attributed to MVIMS, which makes it difficult to predict specific CPU usage averages. For example, an MVIMS-monitored BMP that issues many simple DL/I calls and does little other processing (such as a test DFSDDLT0) shows a much higher MVIMS percentage than an application program that does more internal processing (which causes little MVIMS work).

- The IMS workload profile can have a major effect also because it, too, can vary the amount of work that the Event Collector does in relation to IMS. This variation includes such things as the ratio of data communications work to database work, the type of database DL/I calls being made (simple or complex, GET or UPDATE), and the number of database I/Os per DL/I call. In addition, many IMS systems do not have a consistent workload profile throughout the day or from one day to the next, making valid comparisons extremely difficult.

MVIMS Event Collector User Exits

There are three user exits that can be invoked from the Event Collector. Control is given to a routine at

- end of first DB and first DC DL/I call (IMRUDLI exit)
- end of transaction processing (IMRUTRN exit)
- end of program processing (IMRUPGM exit)

Reporting IMS Transactions to the SRM

MVIMS can report data for all IMS transactions to the OS/390 System Resource Manager (SRM).

Note: When OS/390 is operating in Goal Mode, MVIMS does not report transaction events to SRM.

To display the report data, request a CMF or RMF Workload Activity report. The basic data that MVIMS collects on a per-transaction basis is sufficient to allow CMF and RMF to calculate average transaction-ended (response) time.

Note: Capacity Management Facilities (CMF) is a BMC Software product, and Resource Management Facility (RMF) is an IBM product.

The parameters for the OS/390 installation control specifications (ICS) are located in the IEAICS*xx* member of the SYS1.PARMLIB data set.

To activate the data collection, specify the SUBSYS and TRXCLASS parameters to both OS/390 and MVIMS.

SUBSYS=*xxxx*

The value *xxxx* is a one- to four-character subsystem name. You must also specify the same subsystem name with a SUBSYS parameter in the IMFECP00 member of the BBPARM data set, and the value must be identical to the value specified for the SUBSYS parameter in SYS1.PARMLIB member IEAICS*xx*. This parameter specifies that IMS transactions are to be reported to the OS/390 System Resource Manager (SRM) under this subsystem name. MVIMS uses the SYSEVENT facility.

Note: If the SUBSYS parameter is omitted from the SYS1.PARMLIB member IEAICS*xx*, the default is no IMS transaction reporting. If the SUBSYS values in BBPARM member IMFECP00 and SYS1.PARMLIB member IEAICS*xx* do not match, the data is not collected.

TRXCLASS=CMF*nnnn*,RPGN=*rrr*

CMF is fixed, and *nnnn* represents a valid control performance group number (PGN) in the active message processing region (MPR).

The value *rrr* represents the report performance group number (RPGN) assigned to this TRXCLASS entry. For all transactions that are reported under the corresponding SUBSYS and TRXCLASS, SRM accumulates transaction data in the specified report performance group.

Figure 7-1 provides an example of parameter specifications in SYS1.PARMLIB member IEAICS*xx*. MVIMS requires the parameter specifications to report IMS transactions to the OS/390 System Resource Manager (SRM).

Figure 7-1 Sample Parameter Specifications for Reporting IMS Transactions to OS/390

```
SUBSYS=STC, PGN=3              /* Started tasks    */
 TRXNAME=IMS,PGN=4             /* IMS control rgn  */
 TRXNAME=IMSMSG01,PGN=5        /* IMS msg region 1 */
 TRXNAME=IMSMSG02,PGN=5        /* IMS msg region 2 */
 TRXNAME=IMSMSG03,PGN=5        /* IMS msg region 3 */
 TRXNAME=IMSBMP01,PGN=6        /* IMS bmp region 1 */
 TRXNAME=IMSBMP02,PGN=6        /* IMS bmp region 2 */
```

Parameters like those shown in Figure 7-1 would probably already exist in ICS. The parameters show that a started task with name IMS is assigned PGN4, IMSMSG tasks are assigned PGN5, and IMSBMP tasks are assigned PGN6. The additional parameters shown in Figure 7-2 must also be specified.

In the example shown in Figure 7-2, all transactions reported under subsystem *xxxx* with a PGN of 5 are accumulated under RPGN 710. In other words, all transactions executed in an active MPR (within ICS STC SUBSYS definition) in PGN5 are accumulated under RPGN 710. Transactions with a PGN of 6 are accumulated under RPGN 720.

Figure 7-2 Additional Specifications Required for Reporting to OS/390

```
SUBSYS=xxxx,RPGN=700
 TRXCLASS=CMF0005,RPGN=710  /* for pgn 5 */
 TRXCLASS=CMF0006,RPGN=720  /* for pgn 6 */
```

Note: Since all transactions are reported from the dependent regions, there is no need to set up an RPGN for the control region (PGN=4) in this example.

For each IMS transaction, MVIMS issues the SYSEVENT macro with the following parameters:

SUBSYS=*xxxx*, TRXCLASS=CMF*nnnn*

The value *nnnn* is the control performance group number (PGN) assigned to the task under which this transaction executed. The PGN is assigned through ICS.

Chapter 8 Customizing with User Exit Routines

This chapter describes the sample user exit routines provided by BMC Software and explains how you can customize the routines to meet your needs.

Event Collector User Exit Routines

The Event Collector user exit routines are skeleton programs in BBSAMP from which you can build more extensive routines.

Transaction Record User Exit (IMRUTRN)

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

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

See "IMRUTRN and IMRUPGM Cross-Memory Mode Considerations" on page 8-2 for special considerations about using this exit.

Program Record User Exit (IMRUPGM)

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

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

See "IMRUTRN and IMRUPGM Cross-Memory Mode Considerations" on page 8-2 below for special considerations about using this exit.

IMRUTRN and IMRUPGM Cross-Memory Mode Considerations

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

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

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

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

The BACKOUT parameter, specified in IMFECP00, has a default of YES (see Table 7-2 on page 7-10).

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

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

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

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.

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

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

Log Edit User Exit Routines

Two user exit routines supplied by BMC Software, PRSLEXIT and PRSCEXIT, provide user access to the IMS log file records and to IRUF data. PRSLEXIT is a dummy exit routine for the IMS log file records. PRSCEXIT is a default exit routine for IRUF data. These routines are written in Assembler language and are distributed in BBSAMP members PRSLEXIT and PRSCEXIT.

Sample routines that can be used for a user-written log exit routine are located in the following BBSAMP members:

PRSLEXTA	Assembler log record exit
PRSLEXTC	COBOL log record exit

Sample routines that can be used for a user-written IRUF exit routine are located in the following BBSAMP members:

PRSCEXIT	Default Assembler exit routine to IRUF data
PRSCEXTC	COBOL exit routine to 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, PRSCEXIT, or both, instead of the routines supplied by BMC Software, you must link the user-written routines with a front-end module (PRSLEXTB or PRSCEXTB), using the link edit statements from BBSAMP members LNKCEXIT and LNKLEXIT.

Program IMFLEDIT calls PRSCEXIT 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 PRSCEXIT 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:

Entry Point	Description
LEXITEP1	for initialization
LEXITEP2	for log record processing
LEXITEP3	for exit termination

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 OS/390 language that supports standard subroutine linkage.

Note: If the exit is written in COBOL, use the RESIDENT and NODYN compile options and the runtime option RTEREUS=YES.

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.

Entry Point	Description
CEXITEP1	for initialization
CEXITEOF	for exit termination
CEXITTER	for terminal accounting record processing
CEXITPSB	for program accounting record processing
CEXITTRN	for transaction accounting record processing

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 OS/390 language if standard subroutine linkage is supported.

Note: If the exit routine is written in COBOL, use the RESIDENT and NODYN compile options and runtime option of RTEREUS=YES.

Summarized IRUF Exit

TASCOSTR supports a user exit that allows you to change the IRUF file during creation. A sample COBOL exit is provided in BBSAMP member TASEXIT. This exit must be named TASEXIT and is link-edited into module TASCOSTR.

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

If the exit is written in COBOL, use the RESIDENT and NODYN compile options and runtime option of RTEREUS=YES.

Exit Invocation Points

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

Entry Point	Entry Time
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, CIMTAR02, CIRUFR02, 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(TACCOSTR)
ENTRY TACCOSTR
NAME TACCOSTR(R)
```

These statements are in BBSAMP member TACCOSTR.

Chapter 9 Optional Modifications

This chapter documents optional modifications to MAINVIEW for IMS.

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 identifier is the SAP transaction code.

This support allows you to analyze your SAP workload 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 the SAP 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 the LINKSAP 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.

SMP-Applied Modifications

The IBM System Modification Program (SMP) should be used to apply the modifications described in the following two sections. Use of SMP ensures that the modifications are not regressed when an MVIMS service is applied. If an MVIMS 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.

```
//         JOB
//         EXEC PGM=AMBLIST
//SYSPRINT DD   SYSOUT=*
//SYSLIB   DD   DSN=&prefix.BBLINK,  <=== BBLINK LIBRARY
//              DISP=SHR
//SYSIN    DD   *
LISTLOAD OUTPUT=XREF,MEMBER=(IMECSAU1,IMECSAU2,IMECSAU8,IMECSAU9)
//*
```

Modification to Support TELON

This modification supports the TELON application package. There are two ways to activate MVIMS support for TELON.

1. Specify TELON=YES in BBPARM member IMFECP00.

   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 MVIMS log records.

   Specifying TELON=YES also makes the information available to the IMFLEDIT user exit routines (see “Log Edit User Exit Routines” on page 8-3).

2. 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.

Load Module	CSECT	Entry Point
IMECSAU*x*	IMEDL3E*x*	TELON1
IMECSAU*x*	IMELTRN*x*	TELON2

   The variable *x* in the load module and CSECT names can be the

   8 suffix for IMS 5.1
   9 suffix for IMS 6.1
   1 suffix for IMS 7.1
   2 suffix for IMS 8.1

   If the MAINVIEW for IMS modules are already copied to the IMS library, the modified version of IMECSAUx must be copied to IMS (see “Modifying the IMS Control Region JCL” on page 6-8).

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 IMFECP00.
- Use the TELON1 ZAP, rather than the TELON2 ZAP.

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.

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

Load Module	CSECT	Entry Point
PRSPARM	PRSPARM	ENMSGFLG

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

X'01' To suppress warning messages issued for BMPs and JBPs only

X'02' To suppress warning messages issued for BMPs, JBPs, MPPs, and JMPs

Chapter 10 Log Edit

The Log Edit component of MVIMS Offline processes transaction workload data that is recorded in the IMS log. This processing prepares the data for use by MVIMS Offline. The Log Edit component consists of

- input data
- Log Edit input parameters
- IMFLEDIT program
- user exit routines
- output reports
- output file (a detail IRUF)

Log Edit Input Data

Input to Log Edit consists of IMS transaction workload data that has been recorded in the IMS log. Some of the IMS log records are written by IMS and some are written by the Event Collector. The Log Edit IMFLEDIT program processes all of the MVIMS records and some of the IMS records to produce reports and files that can serve as input to the MVIMS Offline components. All input data should represent a single day, as described in “Establish Daily Cut-Off Time” on page 2-3.

To implement input record selection through the standard IMS log archive process, specify a user output file and record selection control statements.

With the IMFLEDIT program, you can extract the data directly from the IMS log or from an archived subset of the log. Sample log archive JCL is provided in MVIMS BBSAMP member ARCHJCL. Sample control statements are provided in members ARCHCTL1 and ARCHCTL2.

Tip: To avoid producing a separate data set during each archive run, you can specify DISP=MOD for the user output file.

MVIMS Records Only

To select only MVIMS records for processing by IMFLEDIT, use the following sample control statements from BBSAMP member ARCHCTL1 with the IMS log archive utility:

```
X'F9'
X'FA'
```

If you use only the X'F9' and X'FA' control statements, the IMS log records that are required to collect output queue time, special message switch, and multiple systems coupling (MSC) data will be excluded. The IMFLEDIT output transaction record fields for these items will contain zeros.

MVIMS and IMS Records

To select both the required MVIMS-written records and also the IMS-written records, use the following sample control statements from BBSAMP member ARCHCTL2 with the IMS log archive utility:

```
X'01'    X'31'    X'43'
X'03'    X'33'    X'48'
X'06'    X'34'    X'59'
X'08'    X'35'    X'F9'
X'14'    X'36'    X'FA'
X'15'    X'38'    X'4003'
```

Log Edit Input Parameters

The IMFLEDIT program processes input data according to parameters specified in BBPARM member IMFLEP00.

Table 10-1 on page 10-3 describes the Log Edit input parameters and lists them in alphabetical order. The parameter defaults are underlined in the table.

Note: Leading zeros are required for the parameter values.

Table 10-1 IMFLEP00 Log Edit Parameters (Part 1 of 5)

Parameter	Valid Values	Description
CMPFMT32	NO YES	Specifies the format of IRUF records created by IMFLEDIT. NO IRUF records are created in the MVIMS version 3.3.*xx* format, with a maximum LRECL of 30,970 and a maximum of 228 DBTs in the TAR record. YES IRUF records are created in the MVIMS version 3.2 format, with a maximum LRECL of 10,056 and a maximum of 50 DBTs in the TAR record. **Note:** The differences between the version 3.2 and 3.3.*xx* formats are described on page 10-8.
CPUTIME	TRUNC ROUND	Specifies how CPU time is to be reported. TRUNC Calculates the CPU time to thousandths of a second and truncates it. ROUND Calculates the CPU time by rounding it to the nearest thousandth of a second.
CTLBFR	05 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `CTL-BUFF-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
CTLDLI	06 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `CTL-DLI-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
CTLOC	01 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) when calculating: `CTL-OPEN-CLOSE-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
DAY	ALL FIRST SECOND DATE	Specifies whether MVIMS is to process a portion of the IMS log file using the program and transaction record start timestamps. ALL Processes all records encountered in IMS log file. FIRST Limits IMS log file data processing to data for the first date encountered. SECOND Limits IMS log file data processing to data for the second date encountered. DATE Limits IMS log file data processing to data for the date specified in the SELDATE parameter. If DATE is specified, the next parameter must be SELDATE=yyddd.

Table 10-1 IMFLEP00 Log Edit Parameters (Part 2 of 5)

Parameter	Valid Values	Description
FAUTIL	NO YES	Specifies whether transaction records (X'FA') encountered on the LOGFILE file will be written to the FAUTIL file. NO Transaction records from the LOGFILE file are not written to the FAUTIL file. YES All transaction records encountered in the LOGFILE file are written to the FAUTIL file. Before writing the transaction records, IMFLEDIT adjusts the timestamps in the transaction records so that they remain consistent with the corresponding timestamps in the transaction accounting records (TARs) written to the IRUF (including the timestamp related to MSGSWIQT).
IMSLEVEL	0000 5100 (IMS 5.1) 6100 (IMS 6.1) 7100 (IMS 7.1) 8100 (IMS 8.1)	Identifies the IMS release level of the input log data sets. Although Log Edit can process log data sets for different releases of IMS, it can process only one level at a time. **Note:** If Log Edit does not find either a transaction (X'FA') or a program (X'F9') log record in the first 50,000 records of the log data set *and* if the value of IMSLEVEL is not the correct release level of the log data set, Log Edit stops with an 0C*x* abend.
LTERMREC	ACTIVE ALL	Specifies the terminals that IRUF terminal accounting records (LARs) are created for. ACTIVE IRUF LARs are created for only those terminals with activity against them. ALL IRUF LARs are created for all terminals defined to IMS.
LTERMTBL	001500 000001– 999999	Specifies the number of 102-byte entries for the logical terminal table. This parameter should be adjusted to a number larger than the number of logical terminals defined to IMS.
MAXINQ	00000 00000– 99999	Optional ceiling value in seconds for TAR INPUT Q. If the ceiling is nonzero, all TAR INPUT Q values that exceed the ceiling are replaced by the ceiling value. If the value is 00000, no ceiling is set or checked.
MAXRINQ	00000 00000– 99999	Optional ceiling value in seconds for TAR R INPUT Q. If the ceiling is nonzero, all TAR R INPUT Q values that exceed the ceiling are replaced by the ceiling value. If the value is 00000, no ceiling is set or checked.
MAXRRESP	00000 00000– 99999	Optional ceiling value in seconds for TAR R RESPONSE. If the ceiling is nonzero, all TAR R RESPONSE values that exceed the ceiling are replaced by the ceiling value. If the value is 00000, no ceiling is set or checked.
MAXSEGS	075 000– 225	Specifies the maximum number of DBT segments to be included in the TAR DBT section. **Note:** The PCB segment is not included in this value; it resides in the fixed portion of the TAR record. The ALLDBS, DB2, and OTHER segments are in addition to the number of segments specified with MAXSEGS.

Table 10-1 IMFLEP00 Log Edit Parameters (Part 3 of 5)

Parameter	Valid Values	Description
MSCTBL	00000 00000– 999999	Specifies the number of 44-byte entries for the multiple systems coupling (MSC) table. If a nonzero value is specified for MSCTBL, IMFLEDIT collects the transaction arrival timestamp in the *processing* system to calculate the input queue time and the response time. The default is 00000 (no MSC handling). If you have MSC processing transactions in your system and the overflow value in the IMFLEDIT Audit Report is not zero, increase the MSCTBL value by the overflow value. **Note:** In MVIMS log records created for MSC transactions, the transaction arrival timestamp is set to the transaction start time, which results in zero input queue time. This processing is done because the arrival timestamp set by IMS is from the originating system, not from the processing system. Since the clocks on the two systems may not be synchronized, the timestamp might be invalid and could lead to anomalies such as a negative input queue time.
MSGBFR	10 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `MSG-BUFF-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
MSGCPU	30 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `MSG-REGION-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
MSGDB2	18 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `MSG-DB2-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
MSGDLI	18 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) as the percentage when calculating: `MSG-DLI-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.

Table 10-1 IMFLEP00 Log Edit Parameters (Part 4 of 5)

Parameter	Valid Values	Description
MSGOC	01 01– 99 REJECT	Indicates to Log Edit what to do when an invalid CPU timing field is encountered. • Use the specified numeric value (01– 99) when calculating: `MSG-OPEN-CLOSE-CPU-TIME = Transaction Elapsed Time × Percentage` For more information, see the note on page 10-7. • REJECT the input log record.
MSGSWIQT	NO YES	Specifies whether to use the input queue time of the originating message or the input queue time of the most recent message switch, which is the time when the message is enqueued to the scheduler message block (SMB). NO Input queue time of the originating message is used. YES Input queue time of the most recent message switch is used.
MSGSWTBL	01000 00001– 999999	Specifies the number of 36-byte entries for the message switch table. If the overflow value in the IMFLEDIT Audit Report is not zero, increase the MSGSWTBL value by the overflow value.
PSTNUM	031 001– 999	Specifies the maximum number of PSTs.
RSPOPT	YES NO	Specifies whether response time data is extracted from the IMS log file. YES Performance Reporter response reports (PRSRESP) are produced from the IRUF. Response times that include the output queue time made available by RSPOPT can be shown on several Performance Reporter reports, such as PRSTRN10. NO Performance Reporter response reports (PRSRESP) cannot be produced from the IRUF. Response times that include the output queue time made available by RSPOPT cannot be shown on several Performance Reporter reports, such as PRSTRN10. These response times are set equal to input queue plus elapsed times.
RSPPURGE	120 000– 999	**Used only if parameter RSPOPT=YES.** Specifies the maximum time in seconds that a response entry will be held in the response table waiting for either the first attempt to transmit or the final dequeue of the message. If the maximum time is exceeded, the missing timestamps are set using this interval and the entry is purged. **Note:** If RSPPURGE is increased, you may need to increase the required size of the response table.
RSPTBL	00500 00001– 999999	Specifies the number of 120-byte entries for the response table. If the overflow value in the IMFLEDIT Audit Report is not zero, increase the RSPTBL value by the overflow value. **Note:** If you specify the maximum value (999999) for this parameter and for LTERMTBL, MSGSWTBL, and MSCTBL, you will allocate approximately 272 MB of storage. If that results in an 878 abend, increase the region size for the job or reduce the specifications.
SELDATE	yyddd	Specifies the Julian date for IRUF record selection. This date is required if DAY=DATE is specified. If not required, this parameter should be commented out, deleted, or left with an unused but valid date to prevent syntax errors. **Note:** An error will also occur if a nonnumeric value is entered (NULL or blanks, for example).

Table 10-1 IMFLEP00 Log Edit Parameters (Part 5 of 5)

Parameter	Valid Values	Description
SELREC	01–FF	Optionally identifies the type of log record to be written to the RTAUDIT file. If SELREC is not used, no records are written to RTAUDIT. Only one type per run can be selected. **Note:** The SELREC function is not available if SMQ=YES (page 10-7) and RSPOPT=YES (page 10-6) are specified. If you need to write records from a shared queues group of log data sets to the RTAUDIT file, set SMQ=YES and RSPOPT=NO. Keep in mind, however, that the generated IRUF TARs will not be valid.
SMQ	NO YES	Specifies whether special IMFLEDIT processing will be done for shared queues groups. NO No special IMFLEDIT processing is done for shared queues groups. YES The logs from all IMS systems participating in a shared queues group must be concatenated to the LOGFILE DD, and the resulting IRUF will have TARs, LARs, and PARs for all participating IMS systems. **Note:** The SELREC function (page 10-7) is not available if SMQ=YES and RSPOPT=YES (page 10-6) are specified. If you need to write records from a shared queues group of log data sets to the RTAUDIT file, set SMQ=YES and RSPOPT=NO. Keep in mind, however, that the generated IRUF TARs will not be valid. If SMQ=NO, the SMQ GRP NM field in the IMFLEDIT Audit Report will be blank. If SMQ=YES, the SMQ GRP NM field will contain the name of the shared message queue group.
SYSID	Blank	Not used.
TIMSTAMP	ACCEPT REJECT	Indicates to Log Edit what to do when input records with invalid arrival, start, or stop timestamps are encountered. ACCEPT Log Edit accepts the record but uses default timestamps. REJECT Log Edit rejects the record and lists it in the Log Edit Error Report.

Note: IMFLEDIT automatically adjusts invalid CPU timing fields by using the specified numeric value and multiplying it by transaction elapsed time. For example, if the numerical value for MSGCPU is 30 and the transaction elapsed time is 1 second, IMFLEDIT automatically adjusts it to 0.3 seconds:

```
MSG-REGION-CPU-TIME=(1 × 0.30)
```

CMPFMT32 Format Option

This format option is provided as a migration path from MVIMS version 3.2. You should use this option if you have written programs that process the IRUF files. The option will be discontinued in the next product release. Specify **CMPFMT32=YES** to write records in the version 3.2 format.

Differences between Version 3.2 and Version 3.3.*xx* IRUF Formats

Table 10-2 highlights the differences between the two IRUF formats.

Table 10-2 IRUF Record Formats for MVIMS Versions 3.2 and 3.3.*xx*

3.3.*xx* IRUF Format	3.2 IRUF Format
Fixed portion of TAR record is 676 bytes long and includes the PCB segment.	Fixed portion of TAR record is 456 bytes long and does not include the PCB segment.
Maximum number of DBT segments is 225. The maximum number is controlled by the IMFLEDIT MAXSEGS parameter, which has a default value of 75. Up to three additional DBT segments (ALLDBS, DB2, and OTHERS) will be included in the TAR IRUF DBT if they were found in the X'FA' record DBT. If an OTHERS segment was not found in the X'FA' DBT and the MAXSEGS value is exceeded, an OTHERS segment will be created.	Maximum number of DBT segments is 50. Processing of the X'FA' record DBT segments stops immediately after the fiftieth output DBT segment has been created in the IRUF. Statistics for databases in excess of 50 DBT segments are lost.
TAR DBT section contains DBD, DB2, and RESP segments. The PCB segment is in the TAR IRUF fixed segment and is 192 bytes long.	TAR DBT section contains PCB, DBD, DB2, and RESP segments.
Each segment is 132 bytes long.	Each segment is 192 bytes long.
DCB attributes are: DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974)	DCB attributes are: DCB=(RECFM=VBS,LRECL=10056,BLKSIZE=13030)

TAR Segment Processing for Version 3.3.*xx* IRUF Format

With MVIMS version 3.3.00, the maximum number of DBT segments allowed in the TAR IRUF DBT was increased from 50 to 228, and the way segments are processed was changed.

First, all DBT segments in the X'FA' record are processed. Then, if the MAXSEGS value is exceeded (see page 10-4), the most "significant" segments are identified and are written to the TAR IRUF trailer individually. The counts for the remaining segments are combined in the OTHERS segment and are written to the IRUF trailer.

The significance of a segment is determined by the total of the DL/I and I/O counts for each DB-DBT segment.

If ddname information was requested through the IMFECP00 member, the counts for the ddname segment are added to the related DB-DBT segment counts for determination of significance. When the TAR record is created, the primary DB-DBT segments are immediately followed by their related DD-DBT (ddname) segments.

IMFLEDIT Execution

To execute IMFLEDIT, use the JCL statements in Table 10-3. (Sample JCL is provided in Figure 10-1 on page 10-13.)

Table 10-3 IMFLEDIT JCL Statements (Part 1 of 3)

Statement	Use
EXEC	Specifies the name of the Log Edit program (PGM=IMFLEDIT). The region requirement could be affected by the • log file block size • IRUF block size • number of buffers specified for the data sets • size of the GETMAIN area in the PRSRSPED subroutine • internal table sizes (LTERM, response, message switch, multiple systems coupling) as specified by the IMFLEP00 input parameters **Note:** You can add the PARM='*nn*' parameter to the EXEC statement to designate an alternate member name. For example, to designate IMFLEP01, you would include PARM='01' in the EXEC statement. If the '*nn*' value is not two characters long, the 00 default value is used. If the resulting member is not found, IMFLEDIT stops with a return code of 32.
STEPLIB DD	Defines the program library containing the IMFLEDIT program load module.

Table 10-3 IMFLEDIT JCL Statements (Part 2 of 3)

Statement	Use
SORTLIB DD	Defines the sort program library (such as DFSORT). **Note:** This statement is used only when IMFLEDIT is being run for shared message queues (SMQ=YES, page 10-7).
LOGFILE DD	Defines IMS log data set to be processed by program IMFLEDIT. If several log files are concatenated, they must be in chronological order, unless IMFLEDIT is being run for shared message queues. When IMFLEDIT is being run for shared message queues (SMQ=YES in IMFLEP00), logs from all IMS systems are concatenated to the LOGFILE DD. When SMQ=YES, logs do not have to be concatenated in chronological order, but the execution will perform better if they are. If the log is in VBS format, specify the following: `DCB=RECFM=VBS`
RESUTIL DD	Defines the IRUF to be created. The required attributes are described below. If CMPFMT32=YES (page 10-3): `RECFM=VBS,LRECL=10056,BLKSIZE=`(see note) **Note:** The block size must be specified explicitly and should not be larger than 500 bytes below the maximum allowed for the device type. For a 3400 series tape drive, the block size should be 32000-500=31500. If CMPFMT32=NO (see page 10-3): `RECFM=VBS,LRECL=30970,BLKSIZE=30974`
AUDITS DD	Defines the message data set containing the Log Edit Audit Report. The following attributes are required: `RECFM=FBA,LRECL=133,BLKSIZE=`(multiple of 133)
RTAUDIT DD	If the SELREC input parameter (described on page 10-7) is specified, this statement defines the data set to which selected records are written. The following attributes are required: `RECFM=VB,LRECL=2400,BLKSIZE=4524`
FAUTIL DD	Contains all transaction (X'FA') records encountered in the LOGFILE. Before writing the transaction records, IMFLEDIT adjusts the timestamps in the transaction records so that they remain consistent with the corresponding timestamps in the transaction accounting records (TARs) written to the IRUF (including the timestamp related to MSGSWIQT). The following attributes are required: `RECFM=VB,LRECL=2400,BLKSIZE=4524` **Note:** This statement is used only when FAUTIL=YES (see page 10-4).
IMFPARM DD	Defines the MVIMS parameter (BBPARM) library. The following attributes are required: `RECFM=FB,LRECL=80,BLKSIZE=`(multiple of 80) The input parameters are specified in the IMFLEP00 member of BBPARM. The valid values, defaults, and effects of the parameters are described in "Log Edit Input Parameters" on page 10-2.

Table 10-3 IMFLEDIT JCL Statements (Part 3 of 3)

Statement	Use
ERRORS DD	Defines the data set containing the Log Edit Error Report. The following attributes are required: `RECFM=FBA,LRECL=133,BLKSIZE=`(multiple of 133)
SYSOUT DD	Defines the data set where DFSORT writes its error messages and performance statistics. **Note:** This statement is used only when IMFLEDIT is being run for shared message queues (SMQ=YES, page 10-7).
SORTDIAG DD	Specifies that the IMFLEDIT internal call to DFSORT will cause the sorting performance statistics to be printed on the SYSOUT DD. The statistics are useful for tuning SORTWK01, SORTWK02, and SORTWK03 to improve DFSORT performance. **Note:** This statement is used only when IMFLEDIT is being run for shared message queues (SMQ=YES, page 10-7).
SORTWK01 SORTWK02 SORTWK03	Define a work data set for sorting shared message queue logs when SMQ=YES is set in IMFLEP00. Note that space should always be allocated in cylinders and that no secondary allocation should be used. These statements should be put on different channel paths, and they should be allocated on the same type of device.

Figure 10-1 on page 10-12 provides IMFLEDIT JCL examples.

Figure 10-1 IMFLEDIT JCL Examples

If CMPFMT32=YES (see page 10-3):

```
//JOBNAME  JOB  ..............
//STEP1    EXEC PGM=IMFLEDIT,REGION=4096K
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SORTLIB  DD   DSN=&SORTLIB,DISP=SHR
//LOGFILE  DD   DSN=IMSLOG,VOL=SER=......,DISP=OLD,UNIT=tape
//RESUTIL  DD   DSN=IRUF.DAYS(+1),DISP=(NEW,CATLG),UNIT=tape,
//              DCB=(RECFM=VBS,LRECL=10056,BLKSIZE=13000,BUFNO=100)
//AUDITS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//RTAUDIT  DD   DSN=XO7REC,DISP=(NEW,CATLG),UNIT=SYSDA,
//              SPACE=(CYL,(2,1)),
//              DCB=(RECFM=VB,LRECL=2400,BLKSIZE=4524)
//IMFPARM  DD   DSN=IMF.BBPARM,DISP=SHR
//ERRORS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD   SYSOUT=A
```

If CMPFMT32=NO (see page 10-3):

```
//JOBNAME  JOB  ..............
//STEP1    EXEC PGM=IMFLEDIT,REGION=4096K
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SORTLIB  DD   DSN=&SORTLIB,DISP=SHR
//LOGFILE  DD   DSN=IMSLOG,VOL=SER=......,DISP=OLD,UNIT=tape
//RESUTIL  DD   DSN=IRUF.DAYS(+1),DISP=(NEW,CATLG),UNIT=tape,
//              DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974,BUFNO=100)
//AUDITS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//RTAUDIT  DD   DSN=XO7REC,DISP=(NEW,CATLG),UNIT=SYSDA,
//              SPACE=(CYL,(2,1)),
//              DCB=(RECFM=VB,LRECL=2400,BLKSIZE=4524)
//IMFPARM  DD   DSN=IMF.BBPARM,DISP=SHR
//ERRORS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD   SYSOUT=A
```

If SMQ=YES (see page 10-7):

```
//JOBNAME  JOB  ..............
//STEP1    EXEC PGM=IMFLEDIT,REGION=4096K
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SORTLIB  DD   DSN=&SORTLIB,DISP=SHR
//LOGFILE  DD   DSN=IMSLOG,VOL=SER=......,DISP=OLD,UNIT=tape
//RESUTIL  DD   DSN=IRUF.DAYS(+1),DISP=(NEW,CATLG),UNIT=tape,
//              DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974,BUFNO=100)
//AUDITS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//IMFPARM  DD   DSN=IMF.BBPARM,DISP=SHR
//ERRORS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD   SYSOUT=A
//SYSOUT   DD   SYSOUT=*
//SORTDIAG DD   DUMMY
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(50))
```

Figure 10-1 IMFLEDIT JCL Examples (continued)

If FAUTIL=YES (see page 10-4):

```
//JOBNAME  JOB  ..............
//STEP1    EXEC PGM=IMFLEDIT,REGION=4096K
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SORTLIB  DD   DSN=&SORTLIB,DISP=SHR
//LOGFILE  DD   DSN=IMSLOG,VOL=SER=......,DISP=OLD,UNIT=tape
//RESUTIL  DD   DSN=IRUF.DAYS(+1),DISP=(NEW,CATLG),UNIT=tape,
//              DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974,BUFNO=100)
//AUDITS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//RTAUDIT  DD   DSN=XO7REC,DISP=(NEW,CATLG),UNIT=SYSDA,
//              SPACE=(CYL,(2,1)),
//              DCB=(RECFM=VB,LRECL=2400,BLKSIZE=4524)
//FAUTIL   DD   DSN=&INDEX..IMF.FAUTIL,DISP=(NEW,CATLG),
//              UNIT=&DASUNIT,SPACE=(TRK,(5,1)),
//              DCB=(RECFM=VB,LRECL=2400,BLKSIZE=4524)
//IMFPARM  DD   DSN=IMF.BBPARM,DISP=SHR
//ERRORS   DD   SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=1330)
//SYSUDUMP DD   SYSOUT=A
```

Note: The RTAUDIT DD statement is required only if the IMFLEP00 log record select option (SELREC) is specified.

Exit Routines

Two supplied user exit routines, PRSLEXIT and PRSCEXIT, provide user access to the IMS log file records and to IRUF data. PRSLEXIT is a dummy exit routine for the IMS log file records. PRSCEXIT is a default exit routine for IRUF data. These routines are written in Assembler language and are distributed in BBSAMP members PRSLEXIT and PRSCEXIT.

For more information about the routines, see “Log Edit User Exit Routines” on page 8-3.

Log Edit Reports

Log Edit produces the following reports:

- Log Edit Audit Report
- Log Edit Error Report
- Parameter File Error Report

Log Edit Audit Report

The Log Edit Audit Report is shown in Figure 10-2 on page 10-15 and Figure 10-3 on page 10-17. The report is a two-page printout of the Log Edit process. The first page provides

- a count of the input records (by record type) in the IMS log file
- a count of the number of records (by record type) sent to the IRUF
- a count of the number of records (by record type) not sent to the IRUF
- a count of the DBT segments merged (by segment type)
- the time span of the data encountered on the input file

Figure 10-2 Log Edit Audit Report, Page 1

```
**** IMF ****                                          IMF LOG EDIT                                  **** IMF ****
CURRENT DATE -  03/22/yy                               AUDIT REPORT                                        PAGE 1

  ************************************************************************
  *                                                *                     *
  *  FILE / RECORD TYPE  <1>                       *  RECORD COUNT <2>*
  *                                                *                     *
  ************************************************************************
  *                                                *                     *
  *  INPUT  FILE                                   *                     *
  *                                                *                     *
  *    NUMBER OF RECORDS READ                      *      4,082,834      *
  *    TRANSACTION ACCOUNTING RECORDS              *        193,213      *
  *    PROGRAM ACCOUNTING RECORDS                  *        146,144      *
  *    IMS START UP RECORDS                        *              0      *
  *    IMS SHUT DOWN RECORDS                       *              0      *
  *    TERMINAL LOGON RECORDS                      *              0      *
  *    TERMINAL LOGOFF RECORDS                     *              0      *
  *                                                *                     *
  ************************************************************************
  *  IMS RESOURCE UTILIZATION FILE                 *                     *
  *                                                *                     *
  *    TRANSACTION ACCOUNTING RECORDS              *        193,213      *
  *    PROGRAM ACCOUNTING RECORDS                  *        146,144      *
  *    TERMINAL ACCOUNTING RECORDS                 *          1,939      *
  *                                                *                     *
  ************************************************************************
  *                                                *                     *
  *  RECORDS NOT WRITTEN TO IRUF                   *                     *
  *                                                *                     *
  *    TRANSACTION ACCOUNTING RECORDS              *              0      *
  *    PROGRAM ACCOUNTING RECORDS                  *              0      *
  *    TERMINAL ACCOUNTING RECORDS                 *              0      *
  *                                                *                     *
  ************************************************************************
  *                                                *                     *
  *  DBT SEGMENTS MERGE STATISTICS                 *                     *
  *                                                *                     *
  *    NUMBER OF DB-DBT SEGMENTS MERGED            *        248,258      *
  *    NUMBER OF DD-DBT SEGMENTS MERGED            *              0      *
  *    NUMBER OF OVERFLOW DB-DBT SEGMENTS MERGED   *              0      *
  *    NUMBER OF OVERFLOW DD-DBT SEGMENTS MERGED   *              0      *
  *    NUMBER OF DBT SEGMENTS MERGED INTO OTHERS   *         73,562      *
  *                                                *                     *
  ************************************************************************
  *                                                *                     *
  *                                                                      *
  *  RANGE OF DATA ENCOUNTERED           IMSID-I8    SMFID-SYSC          *
  *            <3>                          <4>         <5>              *
  *                                                                      *
  *                                                                      *
  *                            SELECTED           BYPASSED               *
  *                            --------           --------               *
  *  EARLIEST ARRIVAL          yy.185 11:00:41                           *
  *  EARLIEST START            yy.185 11:00:41                           *
  *  LATEST STOP               yy.185 12:50:42                           *
  *                                                                      *
  ************************************************************************
```

Table 10-4 describes some of the elements in the first page of the Log Edit Audit Report. The reference numbers (with the **<n>** format) match the elements in the report example pages to the elements described in the table.

Table 10-4 Log Edit Audit Report Elements, Page 1

<1> FILE / RECORD TYPE The name of the record for which the counts are tabulated. The tabulations are categorized as follows: INPUT FILE: The record types processed from the input IMS log file. IMS RESOURCE UTILIZATION FILE: The record types created and written to the IRUF. RECORDS NOT WRITTEN TO IRUF: The MVIMS record types not written to the IRUF because of the DATE parameter selection. DBT SEGMENTS MERGE STATISTICS: The types of DBT segments merged.
<2> RECORD COUNT The number of records of the specified record type processed in this edit run.
<3> RANGE OF DATA ENCOUNTERED The time span of the data read from the IMS log file. The SELECTED column shows the timestamps from the records written to the IRUF. The BYPASSED column shows the timestamps of the bypassed records if the DATE parameter was specified.
<4> IMSID The identification code for the IMS that processed the transactions.
<5> SMFID The identification code from the SMCA for the system that processed the transactions.

The second page of the Log Edit Audit Report (shown in Figure 10-3) provides a statistical report of the edit process as specified by the log edit parameters.

Figure 10-3 Log Edit Audit Report, Page 2

```
**** IMF ****                                        IMF LOG EDIT                                                  **** IMF ****
CURRENT DATE -  03/22/yy                             AUDIT REPORT                                                        PAGE 2

                                    PARAMETER OPTIONS (IMFLEP00) AND PROCESSING STATISTICS
-------------------------------------------------------------------------------------------------------------------------------------
  INPUT    = LOG
  DAY      = ALL
  SELDATE  =             RECORDS NOT SELECTED:              0      TRANS:            0  PROGRAMS:      0        LTERMS:      0
  SYSID    =             RECORDS NOT SELECTED:              0      TRANS:            0  PROGRAMS:      0
  SELREC   =             FB DISPLAY:                        0      FB WARNING:       0  OTHERS:        0
  LTERMREC = ALL         PSTNUM   =                       999      FAUTIL    = NO       MSGSWIQT =   YES
  CMPFMT32 = NO          MAXSEGS  =                       225      SMQ       = YES
  RSPOPT   = YES         IMSLEVEL =                      xx00      SMQ GROUP = DFSXCFGP
  FF ERRORS=       0 <1> FF FA  HIGH/EOF:         0         0  <3> FF DEQ HIGH/EOF:         1          1 <5>
  FP ERRORS=       0 <2> FP FA  HIGH/EOF:         1         1  <4> FP DEQ HIGH/EOF:         2          2 <6>
  RSPPURGE =     999     OUTPUTQ PURGES:           0         0  <7> DEQUEUE PURGES:          0          0 <8> (DEQUEUED AT EOF)
  RSPTBL   =   5,000     HIGH / OVERFLOW:          1         0      UNWRITTEN AT EOF:        0 <9>
  LTERMTBL = 100,000     HIGH / OVERFLOW:        470         0 <10>
  MSGSWTBL =   5,000     HIGH / OVERFLOW:          0         0 <11>
  MSCTBL   =   5,000     HIGH / OVERFLOW:          0         0 <12>
  MSGCPU   =      30     NUMBER OF ERRORS DETECTED:          0      MAXINQ   =         0   NUMBER OF ERRORS DETECTED:           0
  CTLDLI   =       6     NUMBER OF ERRORS DETECTED:          0      MAXRINQ  =         0   NUMBER OF ERRORS DETECTED:           0
  CTLBFR   =       5     NUMBER OF ERRORS DETECTED:          0      MAXRRESP =         0   NUMBER OF ERRORS DETECTED:           0
  MSGDB2   =       6     NUMBER OF ERRORS DETECTED:          0      EXCEPTIONS:        0
  MSGDLI   =      18     NUMBER OF ERRORS DETECTED:          0
  MSGBFR   =      10     NUMBER OF ERRORS DETECTED:          0
  MSGOC    =       1     NUMBER OF ERRORS DETECTED:          0
  CTLOC    =       1     NUMBER OF ERRORS DETECTED:          0
  TIMSTAMP =  REJECT     NUMBER OF ERRORS DETECTED:          0

                                    PARAMETER OPTIONS (IMFECP00)
-------------------------------------------------------------------------------------------------------------------------------------
  BHTO     = OFF         BILLOVHD = YES
  BMP      = YES         CPU      = DEP
  DBIO     = IOWAITS     DEPREC   = NO
  TRNSYNC  = NO          DBTNAME  = DD
  CTLOVHD  = YES
```

IMFLEDIT Performance

If you get an unexpected increase in CPU or processing time when you are running IMFLEDIT, you can use page 2 of the Audit Report and the information in Table 10-5 to find a solution to the problem. If you are not able to solve the problem on your own, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

The reference numbers (with the **<n>** format) match the elements in the report example in Figure 10-3 to the information in Table 10-5 on page 10-18.

Table 10-5 Log Edit Audit Report Elements, Page 2 (Part 1 of 5)

<1> FF ERRORS

A value other than zero in this field indicates a storage problem. The field shows the number of GETMAINs that failed during an attempt to acquire more storage for Full Function X'FA' records.

Solution:

Increase the IMFLEDIT JCL REGION parameter value incrementally by 256 K or more until FF ERRORS is zero.

If you cannot get FF ERRORS to zero, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

<2> FP ERRORS

A value other than zero in this field indicates a storage problem. The field shows the number of GETMAINs that failed during an attempt to acquire more storage for Fast Path X'FA' records.

Solution:

Increase the IMFLEDIT JCL REGION parameter value incrementally by 256 K or more until FP ERRORS is zero.

If you cannot get FP ERRORS to zero, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

<3> FF FA HIGH/EOF

These fields show the highest value and end-of-file value for the number of X'FA' records in the Full Function matching routine. Numbers of 1,000 or more can cause IMFLEDIT performance degradation due to X'FA' record searches. High numbers can be caused by the following circumstances:

- During processing of non-SMQ logs, the transaction name or LTERM name was missing from the X'FA' records in the log.
- During processing of SMQ or non-SMQ logs, IMS was not able to send a response to its destination in a timely manner due to system problems such as a VTAM outage.
- During processing of SMQ logs, the log from the front-end IMS log was not specified in the LOGFILE DD IMFLEDIT JCL statement.
- During processing of SMQ logs, the X'FA' records in the logs did not contain the UOW gathered by the Event Collector, which prevented them from being matched with other log records.
- The matching routine contains a logic error.

Solution:

If you are processing SMQ logs, first check to make sure the front-end IMS log is specified in the LOGFILE DD IMFLEDIT JCL statement. If the LOGFILE DD statement is not the cause of the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

If you are not processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

Table 10-5 Log Edit Audit Report Elements, Page 2 (Part 2 of 5)

<4> FP FA HIGH/EOF

These fields show the highest value and end-of-file value for the number of X'FA' records in the Fast Path matching routine. Numbers of 1,000 or more can cause IMFLEDIT performance degradation due to X'FA' record searches. High numbers can be caused by the following circumstances:

- During processing of non-SMQ logs, the transaction name or LTERM name was missing from the X'FA' records in the log.
- During processing of SMQ or non-SMQ logs, IMS was not able to send a response to its destination in a timely manner due to system problems such as a VTAM outage.
- During processing of SMQ logs, the log from the front-end IMS log was not specified in the LOGFILE DD IMFLEDIT JCL statement.
- During processing of SMQ logs, the X'FA' records in the logs did not contain the UOW gathered by the Event Collector, which prevented them from being matched with other log records.
- The matching routine contains a logic error.

Solution:

If you are processing SMQ logs, first check to make sure the front-end IMS log is specified in the LOGFILE DD IMFLEDIT JCL statement. If the LOGFILE DD statement is not the cause of the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

If you are not processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

<5> FF DEQ HIGH/EOF

These fields show the highest value and end-of-file value for the number of completed responses in the Full Function matching routine. Numbers of 1,000 or more can cause IMFLEDIT performance degradation due to response record searches. High numbers can be caused by the following circumstances:

- During processing of SMQ logs, the log from the back-end IMS log was not specified in the LOGFILE DD IMFLEDIT JCL statement.
- The matching routine contains a logic error.

Solution:

If you are processing SMQ logs, first check to make sure the back-end IMS log is specified in the LOGFILE DD IMFLEDIT JCL statement. If the LOGFILE DD statement is not the cause of the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

If you are not processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

<6> FP DEQ HIGH/EOF

These fields show the highest value and end-of-file value for the number of completed responses in the Fast Path matching routine. Numbers of 1,000 or more can cause IMFLEDIT performance degradation due to response record searches. High numbers can be caused by the following circumstances:

- During processing of SMQ logs, the log from the back-end IMS log was not specified in the LOGFILE DD IMFLEDIT JCL statement.
- The matching routine contains a logic error.

Solution:

If you are processing SMQ logs, first check to make sure the back-end IMS log is specified in the LOGFILE DD IMFLEDIT JCL statement. If the LOGFILE DD statement is not the cause of the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

If you are not processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

Table 10-5 Log Edit Audit Report Elements, Page 2 (Part 3 of 5)

<7> OUTPUTQ PURGES (X'31' purges, during processing and at end-of-file processing) Numbers of 1,000 or more in this field may indicate either a relatively low RSPPURGE parameter value in IMFLEP00 *or* an IMFLEDIT logic error. You can check the OVER column in the PRSRESP report to determine how many X'31' purges have occurred. If transactions have response times in the OVER column, the response times are above the upper limits of the ten time ranges specified in the PRSRESP JCL. If the number of seconds in the OVER column is less than or equal to the number of seconds set in the IMFLEP00 RSPPURGE parameter, all transactions in the OVER column have exceeded the RSPPURGE limit. IMFLEDIT purges an in-flight response when all the transaction log records required to complete the response have not been written by IMS in RSPPURGE seconds. The purges are considered X'31' purges because the X'31' record for a transaction was not encountered within RSPPURGE seconds of the X'03' log record (using IMS log record timestamps). When the response is purged, IMFLEDIT adds the RSPPURGE value to the end time (E-TIME) for the transaction, records the result in R-D-TIME, and writes the result to the IRUF. You can also find evidence of a significant number of X'31' purges by checking the PRSPRINT report R-OUTPTQ time and the PRSRESP report OUTPUT Q and RESP TO DEQ times. **Solution**: If you are not processing SMQ logs, try increasing the RSPPURGE value within tolerable limits to see if the purge numbers decrease. (**Note:** If you set RSPPURGE to a lower value, more responses will be purged before they are completed, resulting in a higher OUTPUTQ PURGES number and a lower RSPTBL HIGH number during processing. If you set RSPPURGE to a higher value, responses will stay in memory longer, and more responses will be completed before they are purged.) If RSPPURGE adjustments do not solve the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23. If you are processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.
<8> DEQUEUE PURGES (X'36' purges, during processing and at end-of-file processing) Numbers of 1,000 or more in this field may indicate either an unusually low RSPPURGE parameter value in IMFLEP00 *or* an IMFLEDIT logic error. You can check the OVER column in the PRSRESP report to determine how many X'36' purges have occurred. If transactions have response times in the OVER column, the response times are above the upper limits of the ten time ranges specified in the PRSRESP JCL. If the number of seconds in the OVER column is less than or equal to the number of seconds set in the IMFLEP00 RSPPURGE parameter, all transactions in the OVER column have exceeded the RSPPURGE limit. IMFLEDIT purges an in-flight response when all the transaction log records required to complete the response have not been written by IMS in RSPPURGE seconds. The purges are considered X'36' purges because the X'36' record for a transaction was not encountered within RSPPURGE seconds of the X'03' log record (using IMS log record timestamps). When the response is purged, IMFLEDIT sets the response time for the transaction equal to RSPPURGE and writes the response time to the IRUF. Another potential reason for the X'36' (output message dequeued) record not being encountered within RSPPURGE seconds of the X'03' record is related to the Communications Manager. The Communications Manager may have sent some part of a multisegment message to a terminal and the user did not clear the screen enough times to allow the last segment of the message to be sent. Or there may have been so many messages queued to so many terminals that the Communications Manager could not complete the delivery and dequeue of some of the messages from the destination queue (written X'36' records) within the RSPPURGE limit. **Solution**: If you are not processing SMQ logs, try increasing the RSPPURGE value within tolerable limits to see if the purge numbers decrease. If RSPPURGE adjustments do not solve the problem, follow the instructions in "IMFLEDIT Problem Support" on page 10-23. If you are processing SMQ logs, follow the instructions in "IMFLEDIT Problem Support" on page 10-23.

Table 10-5 Log Edit Audit Report Elements, Page 2 (Part 4 of 5)

<9> RSPTBL, HIGH / OVERFLOW, UNWRITTEN AT EOF

The RSPTBL HIGH field shows the maximum number of responses stored in the response table at any one time. The OVERFLOW field shows the number of responses that could not be stored and processed due to out-of-storage conditions.

HIGH Problem:

In the HIGH field, numbers of 10,000 or more for non-SMQ logs and numbers of 1,000 or more for SMQ logs can cause IMFLEDIT performance degradation due to response record searches, and such high numbers might indicate an IMFLEDIT logic error.

Solution:

If the HIGH field has numbers of 10,000 or more for non-SMQ logs or numbers of 1,000 or more for SMQ logs, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

OVERFLOW Problem:

If the OVERFLOW field is nonzero, insufficient storage is allocated for the response table.

Solution:

Increase the storage allocated for the response table with the RSPTBL IMFLEP00 parameter until the OVERFLOW field becomes zero. You may also need to change the REGION parameter in the IMFLEDIT JCL EXEC statement to avoid out-of-storage abends.

UNWRITTEN AT EOF Problem:

A number other than zero in this field indicates that a large percentage of responses, relative to HIGH, were not completed and written during EOF processing and might indicate an IMFLEDIT logic error.

Solution:

Try changing the RSPPURGE IMFLEP00 parameter value to an acceptably low limit to check for an improvement in performance and a decrease in response table usage. (**Note:** If you set RSPPURGE to a lower value, more responses will be purged before they are completed, resulting in a higher DEQUEUE PURGES number and a lower RSPTBL HIGH number during processing. If you set RSPPURGE to a higher value, responses will stay in memory longer, and more responses will be completed before they are purged.)

If adjustments to the RSPPURGE parameter do not solve the problem, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

<10> LTERMTBL, HIGH / OVERFLOW

A number other than zero in the LTERMTBL OVERFLOW field indicates that the LTERMTBL IMFLEP00 parameter value is not set high enough to accommodate the number of defined LTERM entries.

Solution:

Increase the LTERMTBL value. You may also need to change the REGION parameter in the IMFLEDIT JCL EXEC statement to avoid out-of-storage abends.

If such adjustments do not solve the problem, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

Table 10-5 Log Edit Audit Report Elements, Page 2 (Part 5 of 5)

<11> MSGSWTBL, HIGH / OVERFLOW

The MSGSWTBL HIGH field shows the maximum number of message-switched responses stored at any one time. The OVERFLOW field shows the number of message-switched responses that could not be stored and processed due to out-of-storage conditions. (**Note:** The RSPPURGE parameter value has no effect on the MSGSWTBL HIGH and OVERFLOW numbers.)

HIGH Problem:

Numbers of 10,000 or more in the HIGH field can cause IMFLEDIT performance degradation due to message-switched response record searches, and such high numbers might indicate an IMFLEDIT logic error.

Solution:

If the HIGH field has numbers of 10,000 or more, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

OVERFLOW Problem:

If the OVERFLOW field is nonzero, insufficient storage is allocated for the message switch table.

Solution:

Increase the storage allocated for the message switch table with the MSGSWTBL IMFLEP00 parameter. You may also need to change the REGION parameter in the IMFLEDIT JCL EXEC statement to avoid out-of-storage abends.

<12> MSCTBL, HIGH / OVERFLOW

The MSCTBL HIGH field shows the maximum number of multiple-systems-coupling (MSC) responses stored at any one time. The OVERFLOW field shows the number of MSC responses that could not be stored and processed due to out-of-storage conditions. (**Note:** The RSPPURGE parameter value has no effect on the MSCTBL HIGH and OVERFLOW numbers.)

HIGH Problem:

Numbers of 10,000 or more in the HIGH field can cause IMFLEDIT performance degradation due to MSC response record searches, and such high numbers might indicate an IMFLEDIT logic error.

Solution:

If the HIGH field has numbers of 10,000 or more, follow the instructions in “IMFLEDIT Problem Support” on page 10-23.

OVERFLOW Problem:

If the OVERFLOW field is nonzero, insufficient storage is allocated for the MSC table.

Solution:

Increase the storage allocated for the MSC table with the MSCTBL IMFLEP00 parameter. You may also need to change the REGION parameter in the IMFLEDIT JCL EXEC statement to avoid out-of-storage abends.

IMFLEDIT Problem Support

If you are unable to solve the problem by modifying IMFLEP00 parameters, contact BMC Software Customer Support, and be prepared to send the following documentation to expedite a solution to the problem:

- the IMFLEDIT JCL jobstream
- the IMFLEP00 BBPARM member
- the set of IMS logs on the LOGFILE data set
- the job log of the problem batch run

Log Edit Error Report

The Log Edit Error Report (shown in Figure 10-4) lists data validity errors detected during the edit of the data generated by the Event Collector. The report serves as a diagnostic aid for problem tracking and resolution. The records that contain the errors presented in the report are written to the IRUF after the errors are corrected. Specify REJECT if you want the error report to be generated without writing error records to the IRUF.

Figure 10-4 Log Edit Error Report

```
CURRENT DATE-  03/22/yy                      IMF LOG EDIT ERROR REPORT                                    PAGE NO. 0001
 ERROR DIAGNOSTIC <1>                              IDENTIFICATION.................  TYPE   REC NO.    DATE/TIME
 PRS233W   ROUTING CODE IS MISSING                 TERM=            TRAN=NK5401PE   TRAN    9,833   yy.081 11.02.05
 PRS214W   TRANSACTION NAME IS MISSING FOR NON-BMP TRANSACTION                <2>      <3>     <4>         <5>
```

Elements of the Log Edit Error Report are described in Table 10-6. The reference numbers (with the **<n>** format) match the elements in the report example pages to the elements described in the table.

Table 10-6 Log Edit Error Report Elements

<1> ERROR DIAGNOSTIC	The message number and the explanation of the data validity error.
<2> IDENTIFICATION	The names and codes relating this data validity error to the applicable LTERM (TERM=), transaction code (TRAN=), or program (PROG=).
<3> TYPE	The MVIMS record type in which this data error was detected, which is either a program record (PROG) or a transaction record (TRAN).
<4> RECORD NUMBER	The placement of the record relative to the first record read into the edit process.
<5> DATE/TIME	The time of day when the program or transaction record was recorded. The date is expressed in Julian format (yyddd) and the time is expressed in hours, minutes, and seconds (hh.mm.ss).

Parameter File Error Report

The Parameter File Error Report lists and describes erroneous parameters and provides a brief explanation of the error. Log Edit prints this report only when it detects an invalid parameter. Each erroneous control statement is printed as follows:

```
IM5226I - CONTROL CARD IS: (control statement image)
```

The statement is followed by one or more error messages. Return code 032 is set and Log Edit stops. Valid parameters are described in “Log Edit Input Parameters” on page 10-2.

Error messages are documented in the MVIMS BBMLIB data set. You can view them with the online message application or you can use ISPF to browse BBMLIB.

Return Codes

This section lists the return codes that show the results of Log Edit execution.

Code	Explanation
032	An invalid parameter was specified in the MVIMS parameter library member IMFLEP00. Log Edit prints the MVIMS Parameter File Error Report and stops.
036	Requested storage was not available during the allocation of internal tables. Increase the region size or correct the table size parameters in IMFLEP00.
040	The number of defined LTERMs exceeded the LTERM table capacity.
041	Either the IMSLEVEL specified does not match the IMS release level recorded in the X'FA' or X'F9' record, or no X'FA' or X'F9' records were found in the input log.
042	The IMS release level of the input log data set could not be determined. Specify the IMSLEVEL parameter explicitly.
043	The input data set referenced by LOGFILE DD contains records that are not supported by this release of MVIMS.
044	The LOGFILE contains data sets that are concatenated out of chronological order, or it contains logs from different IMS systems. IMFLEDIT detects this problem only in IMS 6.1 and later. If you provide logs out of order for earlier versions, incorrect response times may be reported by the Performance Reporter. **Note:** If RSPOPT=YES (page 10-6), the chronological log record sequence check is performed to prevent incorrect response times from being reported by the Performance Reporter. If RSPOPT=NO, IMFLEDIT does not calculate response time data and the check is not performed.
045	SMQ=YES was specified in IMFLEP00, but the log being processed is from IMS 5.1 or earlier.
046	SMQ=YES was specified in IMFLEP00, but the X'FA' records do not have SMQ data. Either this log is not from a shared message queues system or the Event Collector running on that system was MVIMS 3.2 or earlier.
047	SMQ=NO was specified in IMFLEP00, but the X'FA' records have SMQ data.
048	No records were written to the IRUF.
050	SMQ=YES was specified in IMFLEP00, but IMFLEDIT encountered an error calling DFSORT. Add //SORTDIAG DD DUMMY to the IMFLEDIT job step and inspect the messages written to SYSPRINT DD.
051	SMQ=YES was specified in IMFLEP00, but all of the X'FA' records in the LOGFILE DD are not from IMS systems in the same shared queues group. To correct the problem, concatenate to the LOGFILE DD only IMS logs from the same shared queues group.
052	The SELREC option was specified, but the RTAUDIT file could not be opened.
053	The FAUTIL=YES option was specified, but the FAUTIL file could not be opened.

Log Edit may continue to run when it encounters nonfatal abnormal conditions. The following return codes indicate such conditions:

Code	Explanation
064	A response table overflow caused loss of response data. Check the processing statistics in the Log Edit Audit Report for the number of occurrences. If necessary, either decrease the purge time (RSPPURGE) or increase the size of the response table (RSPTBL).
068	A message switch table overflow caused loss of message switch arrival times. Check the processing statistics in the Log Edit Audit Report for the number of occurrences. If necessary, increase the size of the message switch table (MSGSWTBL).
072	A multiple systems coupling table (MSCTBL) overflow caused a loss of MSC transaction arrival times. Check the processing statistics in the Log Edit Audit Report for the number of occurrences. If necessary, increase the size of MSCTBL.
084	More storage was requested by subroutine PRSFPLED than was available in the region. Response fields in some transaction accounting records could be lost. Increase the region size for IMFLEDIT. The number of times this error occurred during IMFLEDIT processing is provided in the Log Edit Audit Report.
088	No log records of type X'33' were found in the IMS logs. Response processing requires X'33' log records. If you provided an extract of the IMS log, make sure that the X'33' log records are included. Refer to the BBSAMP member ARCHCTL2.
092	More storage was requested by subroutine PRSRSPED than was available in the region. Response fields in some transaction accounting records could be lost. Increase the region size for IMFLEDIT. The number of times this error occurred during IMFLEDIT processing is provided in the Log Edit Audit Report.
1*xxx*	The LOGFILE DD OPEN failed (where *xxx* represents the return code from the open failure).
2*xxx*	The LOGFILE DD CLOSE failed (where *xxx* represents the return code from the close failure).

Note: The high-water marks (maximum measured value) reached in the response table (RSPTBL), the message switch table (MSGSWTBL), and the multiple systems coupling table (MSCTBL) and the number of purges that occurred in RSPTBL are monitored and presented in the Log Edit Audit Report when Log Edit ends. Check the high-water marks occasionally to avoid loss of data.

Chapter 11 MVIMS Data and IMS Resource Utilization Files

MVIMS Offline components get their input from a component called the IMS Resource Utilization File (IRUF). IRUFs are sequential data sets in variable blocked spanned (VBS) format. They can be either detail or summarized files. The Performance Reporter and the Transaction Accountant generate their performance and accounting reports using IRUFs as input.

Note: IRUFs can also be used as the input for creating a user-selected summary trace, a feature of the MAINVIEW for IMS Online and MAINVIEW for DBCTL products. (For more information, see BBSAMP member IMFLOGTR.)

IRUFs and the MVIMS transaction accounting system are based on the MVIMS customer ID field. This field associates all resource usage measurements and costs with a specific user. The default value for the customer ID field is the LTERM name. Both the IRUF transaction accounting record (TAR) and the IRUF terminal (LTERM) accounting record (LAR) maintain a record of the customer ID.

MVIMS uses customer ID information for summarization and for charge-out billing. The Transaction Accountant program uses the customer ID information to associate resource usage charges with specific customers. All charges are calculated at the transaction level and are accumulated for each customer category.

Users can gain access to each record processed from the IMS log and to all the IRUF data records through exit routines in the Log Edit process (see “Log Edit User Exit Routines” on page 8-3).

Detail IRUFs

The Log Edit program IMFLEDIT creates detail IRUFs from IMS log data. IMFLEDIT processes program and transaction records written to the IMS log by the Event Collector and also (optionally) selected records written by IMS. IMFLEDIT can use the IMS log directly as input, or a subset of the log can be created and used instead (see “Log Edit Input Data” on page 10-1).

Detail IRUFs contain the following three record types:

- transaction accounting records (TARs), built from Event Collector transaction records written to the log (one record per processed transaction)
- program accounting records (PARs), built from Event Collector program records written to the log (one record per program execution)
- terminal (LTERM) accounting records (LARs), built from IMS start, IMS checkpoint, IMS stop, and Event Collector records, combined with dial-up terminal logon and logoff records (one record per terminal session)

Note: Dial-up terminals are assumed connected from logon to logoff (or until last action if there is no logoff).

Leased lines are assumed connected from IMS startup until shutdown (or until the last system activity if there is no IMS shutdown). VTAM terminals are included in this category.

The attributes of each of these three record types are described in Appendix B, “IRUF Record Layout Descriptions.”

Summarized IRUFs

The TASCOSTR summarization program creates summarized IRUFs from one or more detail IRUFs, one or more summarized IRUFs, or any combination of detail and summarized IRUFs.

For a description of the TASCOSTR program, see Chapter 12, “IRUF Summarization and Cost Accounting (TASCOSTR).”

Summarized IRUF records contain only cumulative information. The three record types (transaction, program, and terminal) contain information for the entire time period being summarized, not just for a single execution. For example, terminal records are summarized by LTERM name within MVIMS customer ID. One record per MVIMS customer ID quantifies all resource usage and reflects all activity that has occurred for an individual terminal in the specified time span. Time span is determined by the data in the input IRUF.

A major benefit of TASCOSTR summarization is that it substantially reduces the volume of records that must be processed, yet retains much of the critical information about transactions, programs, and terminals.

Summarization for each record type is as follows:

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

Note: Because transactions from different users can be processed in one program scheduling, the MVIMS customer ID is not used in program records.

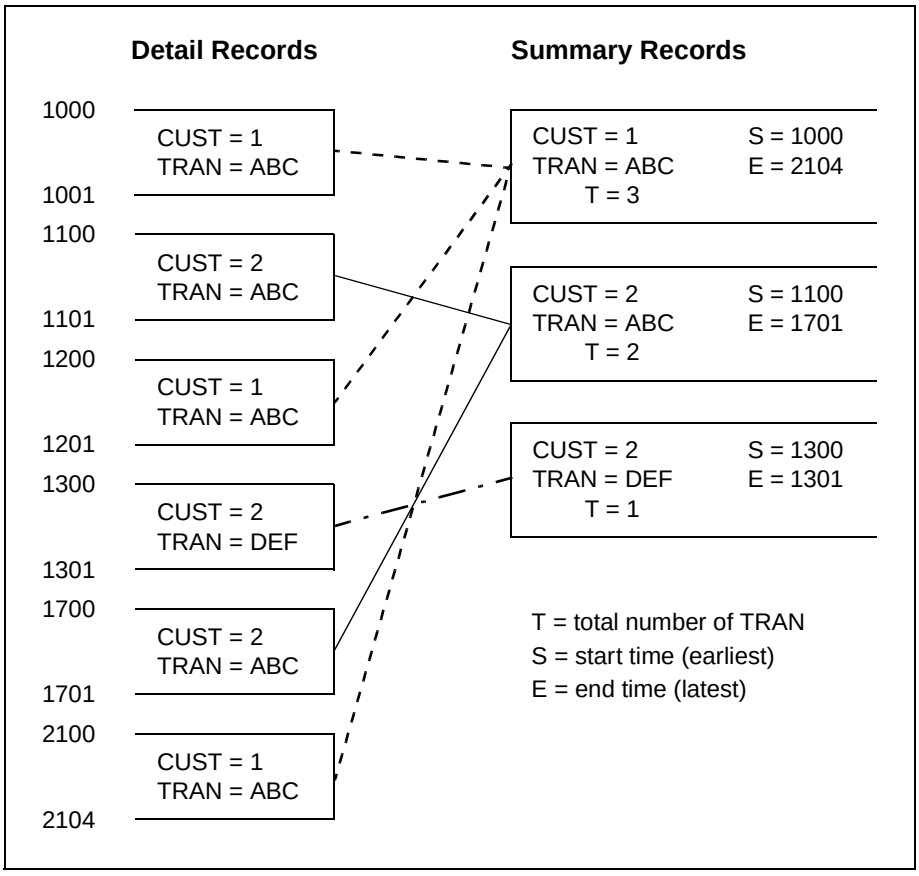
Figure 11-1 on page 11-4 shows how six detail transaction records can be reduced to three summarized transaction records. In this example, two separate customers (Customer 1 and Customer 2) submit transactions to the IMS environment.

Customer 1 submits three transactions (1, 3, and 6) under the same transaction code (ABC). These three transactions are later summarized into a single record for Customer 1. This summarized record contains the MVIMS customer ID and the transaction code ABC. The transaction code indicates that three occurrences of the particular transaction code existed for Customer 1 over the time span of the input data. The *detail* transaction records for these three transactions indicate the specific start and stop time for *each* occurrence of the transaction code. The *summarized* record for this transaction code for Customer 1 identifies the time span in which the transaction code occurred.

For example, the start time in the summarized record indicates the time, within the time span of the input data, of the first occurrence of a particular transaction code for a designated customer; the end time indicates the last time, within the time span of the input data, that the transaction code occurred for this customer.

Similarly, the transactions (2 and 5) submitted by Customer 2 are summarized into a second IRUF record, and transaction occurrence number 4 is summarized into a third IRUF record.

Figure 11-1 Example of IRUF Summarization



IRUF Records

This section describes the three IRUF record types:

- transaction accounting record (TAR)
- program accounting record (PAR)
- terminal (LTERM) accounting record (LAR)

For a description of IRUF record layouts, see Appendix B. For information about how to produce a formatted hardcopy of individual IRUF records, see “Formatted IRUF Record Report” on page 13-18.

All IRUFs can contain either detail or summarized accounting records.

Transaction Accounting Record (TAR)

The detail transaction accounting record (TAR) shows activity for a single execution of a single transaction. The summarized transaction accounting record shows activity for a single transaction code for a single MVIMS customer ID for the summarization time period.

The field structure for detail and summarized records is identical. Summarized records simply combine, in the corresponding field, all individual fields within detail records that can be treated as accumulated values. For values such as core allocated and core used, the summarized record contains the high-water mark.

Each transaction accounting record is divided into two areas, a fixed-data area and an area that contains a variable number of fixed-length segment types (trailers).

Fixed-Data Area

The fixed-data area of a transaction accounting record (TAR) contains the data described in the following sections.

Identification Fields

A TAR contains 28 identification fields. The fields help you associate records with the appropriate customer. They also serve as identifiers in user-written reports. The identification fields are listed alphabetically below. The fields in bold type are used in MVIMS reports.

AGN	PERFORMANCE GROUP
ASID	**PLINE**
CLASS	**PROGRAM**
CUSTOMER ID	**PROGRAM TYPE**
DATABASE NAME	**PTERM**
DATABASE ORG TYPE	**REGION**
DB2 SUBSYSTEM ID	SMFID
FP ROUTE CODE	TERMINAL TYPE
MVIMS **SYSID**	**TRANCODE**
IMSID	**USERID**
LTERM	VTAM NODE
MESSAGE SWITCH DESTINATION	WFI FLAG
MSC DESTINATION ID	SHARED Q UOW
MSC ORIGIN ID	SMQ GROUP NAME

Note: For summarized TAR records, ID fields reflect the values detected in the first occurrence of a transaction code associated with a single customer ID.

Timestamp Information

For a detail TAR, the timestamps show the start time, end time, and other significant times in the life of a single transaction. For a summarized TAR, the timestamps show the start time of the earliest occurrence and the end time of the latest occurrence of a transaction associated with a single customer ID.

Resource Usage Data

A TAR contains information about resource usage, including

- CPU times
- elapsed times
- storage use

Note: In a summarized IRUF, these values are totals or high-water marks.

Terminal Segment (PCB)

This segment contains usage and cost statistics for every logical terminal used for the submission of this transaction. Statistics include message DL/I call counts by type and the number of characters transferred.

Resource Usage Cost Data

A TAR contains information about the costs associated with those areas of resource utilization that are directly attributable to the transaction, including CPU time and storage usage.

Variable Area

The variable area of a transaction accounting record (TAR) contains a variable number of fixed-length segment types (trailers). The segment types are described in the following sections.

Database

This segment type contains usage and cost statistics for each database accessed in processing the transaction. (If the maximum number of trailers is exceeded, the value is shown as OTHERS.) Statistics include database I/O counts and counts of DL/I calls by type.

DB2

This segment type contains usage and cost statistics for a DB2 subsystem if one was accessed in processing the transaction. Statistics include data counts of SQL calls by type. There is one segment for each DB2 plan name.

Response

This segment type contains response statistics that quantify the various response time groupings encountered in processing the transaction. This segment type occurs only in summarized TARs, and only if the required control parameters are specified (see “Response Threshold Definition” on page 12-7).

User-Reserved Fields

A TAR record contains fields reserved specifically for the user. MVIMS does not use these fields for required data. Although some optional data items (not used in MVIMS reporting) are put in one of these areas, the user can overwrite them.

Program Accounting Record (PAR)

The program accounting record (PAR) contains information about the activity of a specific application program. In a detail IRUF, there is one PAR for each program scheduled. In a summarized IRUF, there is one PAR for each program within the summarized time span. This section describes the data contained in each PAR.

Identification Fields

A PAR contains 11 identification fields. The fields can serve as identifiers in user-written reports. The identification fields are listed alphabetically below. The fields in bold type are used in MVIMS reports.

AGN	PERFORMANCE GROUP
ASID	**PROGRAM**
CLASS	**PROGRAM TYPE**
DB2 SUBSYSTEM ID	**REGION**
MVIMS **SYSID**	SMFID
IMSID	

Timestamp Information

In a detail IRUF, the PAR contains the scheduling and program completion time for a single scheduling of a program. In a summarized IRUF, the PAR indicates the start time of the first occurrence of a program and the completion time of the last occurrence of that program.

Resource Usage Data

The PAR contains information about resource usage, including

- scheduling
- elapsed times
- CPU usage
- number of abends

User-Reserved Fields

The PAR record contains fields reserved specifically for the user. The fields are not used by MVIMS.

Note: The PAR contains no MVIMS customer ID field because several transactions originating from different users can be processed in one program scheduling.

Terminal (LTERM) Accounting Record (LAR)

The terminal (LTERM) accounting record (LAR) contains information about the activity of a specific logical terminal (LTERM). In a detail IRUF, there is one LAR per session. In a summarized IRUF, there is one LAR for each logical terminal within an MVIMS customer ID for the entire summary period. This section describes the data contained in each LAR.

Identification Fields

A LAR contains four identification fields. The fields can help you associate records with the appropriate customer. They also serve as identifiers in user-written reports. The identification fields are listed alphabetically below. The fields in bold type are used in MVIMS reports.

CUSTOMER ID
MVIMS SYSID
LTERM
TERMINAL TYPE

Timestamp Information

For dial-up terminals, the timestamp field shows the actual connect time. For leased-line terminals or terminals connected through VTAM, the connect time is set equal to the IMS session time.

Resource Usage Data

The LAR contains information about resource usage, including

- LTERM name
- total connect time for the logical terminal (LTERM)
- number of transactions submitted
- number of characters transferred
- number of message DL/I calls

Note: For a detail record, these values reflect a single connect session. For a summarized record, these values reflect totals of activity for all connect sessions for the logical terminal (LTERM).

User-Reserved Fields

The LAR contains fields reserved specifically for the user. MVIMS does not use the fields.

IMS Elapsed Timings in MVIMS

MVIMS provides nine different timings to define elapsed processing times in the life of an IMS transaction. The Performance Reporter response reports use all nine timings; other Performance Reporter reports use a subset of the timings.

All nine timings are available if the IMFLEDIT response option is in effect during Log Edit (see the RSPOPT parameter on page 10-6). If the response option is not in effect, only the ELAPSED, ARIV START, and ARIV STOP timings are available for each transaction.

Note: The response option does not apply to DBCTL threads.

MVIMS captures response-related elapsed timings (RESPONSE, RSP TO DEQ, OUTPUT Q, DEQUEUE, OUT TO DEQ) only for transactions that respond to the originating terminal and are defined as recoverable to IMS.

In a detail IRUF, the timestamps and elapsed timings define the life of a single transaction. In a summarized IRUF, the timestamps are bounds (earliest arrival or start to latest stop or dequeue time) and the elapsed timings are calculated by dividing the totals by the total number of transactions (or, when calculating response times, by dividing the totals by the total number of response transactions).

Timings

The nine timings provided by MVIMS to define elapsed processing times in the life of an IMS transaction are shown in Table 11-1.

Table 11-1 Timing Definitions

Timing	Definition
RESPONSE	Arrival time to message sent time For message switches, the arrival time of the original input transaction is used.
INPUT Q	Arrival time to start time For message switches, enqueue time is used. The enqueue time depends on the setting of the IMFLEP00 parameter MSGSWIQT. If MSGSWIQT=NO (the default), the input queue time of the originating message is used. If MSGSWIQT=YES, the input queue time of the most recent message switch is used, in other words, the time the message is enqueued to the scheduler message block (SMB).
ELAPSED	Start time to stop time
RSP TO DEQ	Arrival time to message dequeue time For message switches, the arrival time of the original input transaction is used.
OUTPUT Q	Message insert time to message sent time MVIMS measures OUTPUT Q as starting at message insert time, not at transaction stop time. Therefore, RESPONSE is *not* equal to ARIV START plus ELAPSED plus OUTPUT Q.
DEQUEUE	Message sent time to message dequeue time
OUT TO DEQ	Message insert time to message dequeue time
ARIV START	Arrival time to start time For message switches, the arrival time of original input transaction is used. (Original MVIMS input queue time definition.)
ARIV STOP	Arrival time to stop time For message switches, the arrival time of the original input transaction is used. (Original MVIMS response time definition.)

Timestamp Definitions

The nine elapsed timings that can be reported are calculated using the timestamp definitions shown in Table 11-2.

Table 11-2 Timestamp Definitions

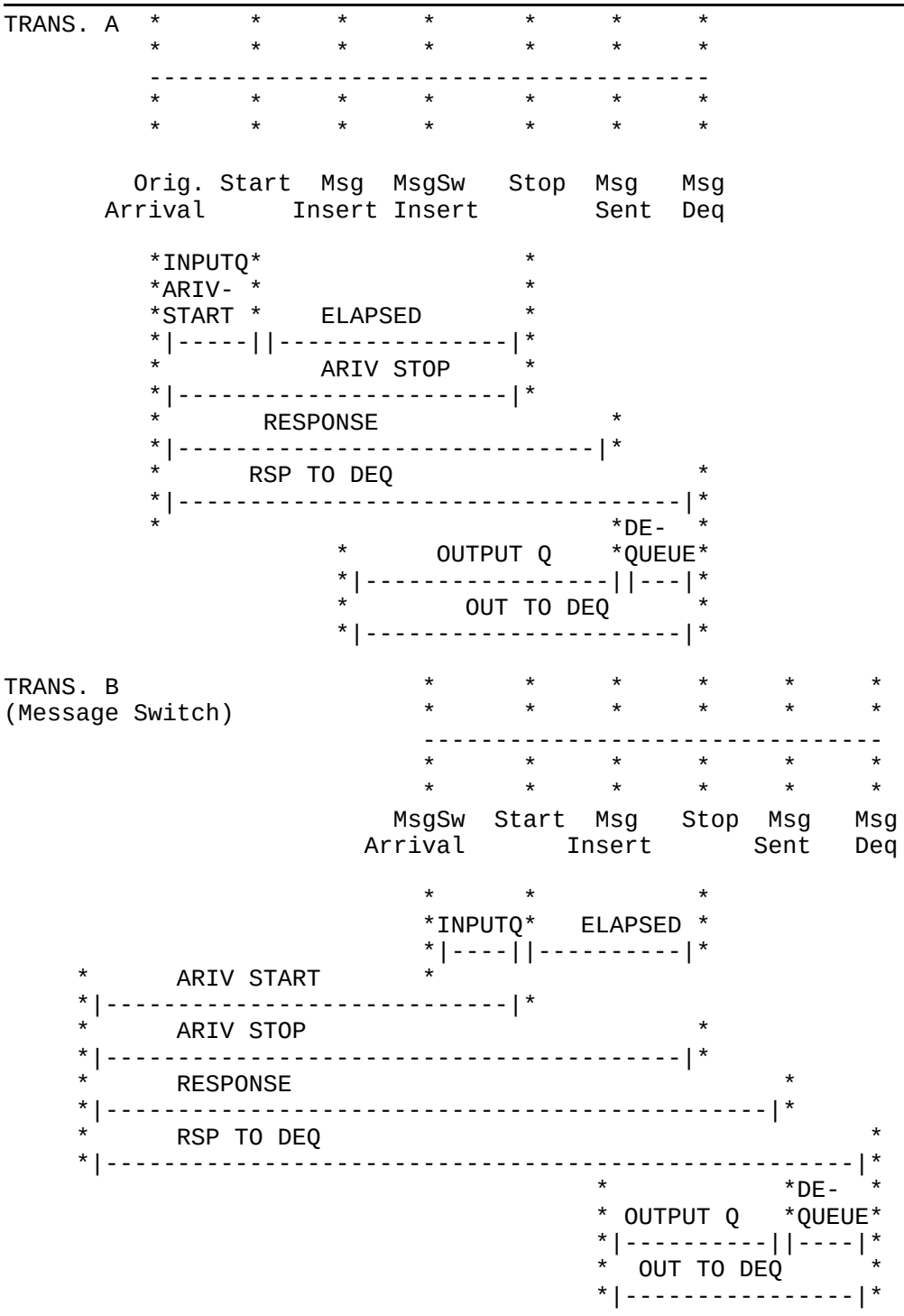
Timestamp	Definition
Arrival Time	Arrival time of the transaction For full function transactions, the arrival time is taken from the X'03' log record, which contains the time the message arrived on the message queue. For Fast Path transactions, the arrival time is taken from the X'FA' record. For message switches, the arrival time of both the message switch transaction and the original input transaction are available.
Start Time	Start of transaction processing The X'FA' record contains the start time for all types of transactions—full function, Fast Path, message switches, and multiple systems coupling (MSC).
Message Insert Time	Enqueue of a response (for summarized IRUFs, enqueue of the first response) to the originating terminal (X'35')
Stop Time	End of transaction processing The X'FA' record contains the stop time for all types of transactions—full function, Fast Path, message switches, and MSC.
Message Sent Time	First attempt to transmit response (communications get unique, X'31') If this time exceeds the maximum specified by the RSPPURGE Log Edit parameter in BBPARM member IMFLEP00, collection stops and the maximum is assigned (see page 10-6). For Fast Path, the 5936 log record is used.
Message Dequeue	Dequeue of the response message (transmission time completed, X'36') If this time exceeds the maximum specified by the RSPPURGE Log Edit parameter in BBPARM member IMFLEP00, collection stops and the maximum is assigned (see page 10-6). For Fast Path, the 5936 log record is used.

Note: A special case concerning elapsed times can occur with some applications, especially with BMPs and JBPs. MVIMS defines transaction elapsed time as the time duration from one MESSAGE GET UNIQUE to the next MESSAGE GET UNIQUE. If a program continues processing after the final MESSAGE GET UNIQUE that received a QC (no more transactions) status code, the resources used during that processing are charged to the last transaction.

The result could be a transaction with a very short elapsed time showing very high resource usage. Such a transaction could cause warning messages to be produced during the Log Edit process if the accumulated CPU time exceeds the transaction elapsed time. Program elapsed time continues through program end.

Figure 11-2 on page 11-15 shows elapsed time examples for a terminal-entered transaction (A) and for a message-switch transaction (B).

Figure 11-2 Timing Examples



IMS CPU Timing in MVIMS

The MVIMS Offline Performance Reporter and Transaction Accountant components report various CPU timing figures for IMS. The timing figures are accumulated by the Event Collector and then maintained in IRUF transaction and program records.

Offline reports show several categories of IMS CPU time. The categories help you analyze CPU time usage patterns. Here are some examples of CPU measurement categories:

- Overhead / Chargeable
  - CPU time that results from system processing
  - CPU time that results directly from a user action
- Application / DL/I / DB2 / Overhead
  - CPU time that the application program accumulates
  - CPU time that IMS DL/I processing accumulates
  - CPU time that DB2 request processing accumulates
  - CPU time that other IMS or OS/390 overhead functions accumulate (such as scheduling, open/close activity, or other system processing)
- Control Region / Message Region
  - CPU time that the IMS control region or DLISAS region accumulates
  - CPU time that is accumulated in the dependent region (message region or batch message region) in which the application was scheduled

Statistics Collected

The IRUF maintains eleven CPU times, eight in the transaction accounting record (TAR) and three in the program accounting record (PAR). The CPU times are described in the section that follows. The eleven CPU times (all in milliseconds) are:

Application program CPU (TAR) – dependent region
Message DL/I CPU (TAR) – dependent region
Control DL/I CPU (TAR) – control region
DB2 CPU (TAR) – dependent region
Message buffer CPU (TAR) – dependent region
Control buffer CPU (TAR) – control region
Message OPEN/CLOSE CPU (TAR) – dependent region
Control OPEN/CLOSE CPU (TAR) – control region
Program scheduling CPU (PAR) – control region
Message region overhead CPU (PAR) – dependent region
Control region overhead (PAR) – control region

The Event Collector CPU data collection parameter specifies which CPU times are to be collected (see page 7-6). If CPU=NONE is specified, all of the above fields, except those showing overhead, are zero. If CPU=DEP is specified, all the above control region fields, except those showing overhead, are zero.

The Performance Reporter and the Transaction Accountant combine this CPU time data in various ways to report on IMS performance and to produce charge-out accounting reports (see the *MAINVIEW for IMS Offline – Performance Reporter Reference Manual* and the *MAINVIEW for IMS Offline – Transaction Accountant Reference Manual*).

CPU Timing Categories

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

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. Usually this time value includes the great majority of the processing of 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 JBPs 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.

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

MVIMS treats control OPEN/CLOSE CPU time as overhead CPU.

This value is collected only 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.

MVIMS treats program scheduling CPU time as overhead CPU.

This value is collected only if CPU=ALL.

Message Region Overhead CPU

The message region overhead CPU value is always collected. The 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. This value generally includes program initialization and termination.

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

- If BILLOVHD=YES, the 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, the 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 this 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 two control region overhead fields:

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

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

Reports That Show CPU Timing Fields

The Performance Reporter and Transaction Accountant reports contain IMS CPU time fields. For a description of the fields, see the report element description sections in the *MAINVIEW for IMS Offline – Transaction Accountant Reference Manual* and the *MAINVIEW for IMS Offline – Performance Reporter Reference Manual*.

The following reports show IMS CPU time data:

- TASCOSTR – Total IMS Resource Usage Analysis report
- PRSPLT00 – Graphical plots of CPU usage
- PRSCLNDR – Transaction Calendar Report
- PRSCLNDR – Program Calendar Report
- PRSCLNDR – System Availability Calendar Report
- PRSCLNDR – CPU Time Calendar Report
- PRSPSBRP – Program (PSB) Processing Report
- PRSPSB20 – Program (PSB20) Processing Report
- PRSTRN10 – Transaction (TRN10) Processing Report
- PRSTRND2 – Transaction (TRND2) Processing Report

Database I/O Data

MVIMS collects all database activity indicators per transaction at the database level. A separate segment is appended to the MVIMS transaction record per accessed database to hold the counts (see the MVIMS transaction log record layout starting on page A-3). This process allows later reporting by transaction, by user (for accounting), and by database, which is always true for the DL/I calls. Each call is counted by type (GU, ISRT, and so on) and per DBD.

These database segments also contain fields for several I/O-related counts. I/O activity is split into categories according to whether the I/O was a read or write, key or nonkey access. One additional count, called NO I/O, is unique to MVIMS. This count is a measure of IMS overhead and is the number of requests to the IMS buffer handler that do *not* result in I/O. These counts can be affected by the Event Collector parameters.

Note: If DBTNAME=DD is specified, database reads and writes and reads without I/O counts (NO I/O counts) are collected at the data set level (except for Fast Path DEDBs).

Database Reads

Reads are counted as key or nonkey reads. No database reads are collected if DBIO=NONE (except for DEDBs and MSDBs).

The database read counts are collected at the database level with both DBIO=BFALTERS and DBIO=IOWAITS.

Reads caused by access through a secondary index or logical database are counted with DBIO=BFALTERS and DBIO=IOWAITS, but are shown for the actual target DBD (DBPCB).

Note: This process may cause key I/O to appear for nonkey databases, such as HDAM.

Database Writes

Writes are counted as key and nonkey writes. No database writes are collected if DBIO=NONE (except for DEDBs).

If DBIO=BFALTERS:

- All writes are collected at the database level.
- Writes to maintain a secondary index or logical database are collected. Counts are maintained at the database level (DBPCB).

If DBIO=IOWAITS:

- Most writes are collected at the transaction level and reported under the database entry ALLDBS, including all writes occurring at sync point (the majority). These counts correspond closely to the DC Monitor IWAITS reported under the I/O PCB (Program I/O report).
- Database writes occurring during call processing (such as the deletion of a HISAM root) are still accumulated by database.
- Writes to maintain a secondary index or logical database are collected under the DBD that is the actual target (DBPCB).
- VSAM-initiated writes for buffer steal and background write are not measured.
- OSAM buffer steal writes are accumulated in the (otherwise unused) NO I/O counter for the database/transaction whose read request forced the buffer steal.
- Chained VSAM or OSAM writes of multiple buffers count as one write.

NO I/O

NO I/O counts are collected only if DBIO=BFALTERS.

NO I/O measures the number of requests to the IMS buffer handler that do not result in I/O.

DB2 Subsystem Activity

MVIMS measures the number of DB2 calls made through the IMS Attach Facility. I/O is not measured. The call counts are recorded in an optional segment at the end of the MVIMS transaction record. The transaction record is described in Appendix B, “IRUF Record Layout Descriptions.”

BMP and JBP Data

BMP and JBP data is always collected unless BMP=NO or BMP=NOCPU is specified. If BMP=NO, no BMP or JBP transaction and program records are produced, which affects all MVIMS 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.

Available Fast Path Data

MVIMS provides the same level of data for Fast Path and mixed-mode workloads as for full function workloads. MVIMS produces fully equivalent transaction and program records for Fast Path with the addition of some data items specific to Fast Path. These records become part of the offline IMS Resource Utilization File (IRUF), which is the input for both the Performance Reporter and the Transaction Accountant. Each record is identified by program type.

Full function workload:

MPP	Message processing and Java message processing (JMP) programs
BMP	Batch message processing and Java batch message processing (JBP) programs
DBT	DBCTL threads (CICS and ODBA)

Fast Path workload:

MDP	Message-driven programs
NDP	Non-message-driven programs
FPU	Fast Path utility

All the following standard types of data are available for Fast Path and mixed mode workloads.

Identification. Trancode, program, region, class, BALG, routing code, databases accessed (including DEDBs and MSDBs), LTERM, VTAM node, user ID.

Timestamps. Program start and end; transaction arrival, start, end, message-sent and message-dequeue.

Elapsed times. Program elapsed; transaction input queue, elapsed, output queue, dequeue, response time.

CPU times. Application program CPU, DL/I CPU, DB2 CPU, overhead CPU, by transaction and by IMS region (dependent, control, DLISAS).

DL/I message calls. All DL/I message calls by type, either through the full function message queues or through the Fast Path expedited message handler, identified with the submitting LTERM and user information.

DL/I database I/O. All DL/I database I/O by type (reads/writes, key/nonkey), to any IMS database, including DEDBs, identified with the database name.

DB2 subsystem access. All SQL requests, which are represented by a plus sign to indicate that they are included in non-Fast Path calls (# NONFP CALLS report field).

In addition, the following Fast Path data items are available per transaction (TAR), program (PAR), or both in the IRUF records:

Program type = MDP, NDP, FPU	(TAR/PAR)
Sync point return code	(TAR)
BALG queue count	(TAR)
NBA = normal buffer allocation	(TAR/PAR)
OBA = overflow buffer allocation	(TAR/PAR)
FP buffers used	(TAR)
FP buffer high-water mark	(PAR)
FP number of CI contentions	(TAR)
FP number of waits for buffers	(TAR)
Routing code	(TAR)
Wait for OBA latch flag	(TAR)
DBD Fast Path organization type = DEDB MSDB nonrelated key in data MSDB nonrelated LTERM key MSDB related fixed MSDB related dynamic	(TAR)

All Performance Reporter and Transaction Accountant reports include the Fast Path transactions, programs, and databases, shown by type.
The Performance Reporter has a particular transaction processing report (PRSTRNFP) that reports the key Fast Path data items, in addition to the standard workload figures. The Total IMS Resource Usage Analysis report provides a profile of Fast Path usage for transactions that use expedited message handling or that access MSDBs and DEDBs.

Chapter 12 IRUF Summarization and Cost Accounting (TASCOSTR)

IMS Resource Utilization Files (IRUFs) are maintained by MVIMS. A detail IRUF is the output created by the MVIMS IMFLEDIT program. This output can later be processed by the MVIMS TASCOSTR program to create a summarized IRUF version. The TASCOSTR program can also process and summarize an IRUF that has been previously summarized. The same rules apply, regardless of whether the input IRUF is a detail or a summarized file. (For a visual overview of the TASCOSTR summarization processes, see Figure 12-1 on page 12-3.)

A summarized IRUF contains information derived from measurements made on a cumulative basis, rather than on a one-to-one basis. When a detail IRUF is processed to create a summarized IRUF, information is summarized so that the three IRUF record types contain information covering the entire time period of the input IRUF, not just a single process or execution.

Input to TASCOSTR can be single or multiple detail IRUFs, single or multiple previously summarized IRUFs, or any combination of detail and summarized IRUFs. However, all input to TASCOSTR must be combined and sorted by customer ID, transaction code, and LTERM before processing.

TASCOSTR provides options that produce accounting information or perform IRUF summarization or both. The summarization option produces the same output as the version delivered with the Performance Reporter. The accounting option produces a series of reports, including the Detail Charge-out Analysis, Distribution Charge-out Analysis, and Total IMS Resource Usage Analysis reports.

When TASCOSTR produces a summarized IRUF, the newly summarized IRUF contains one record for all transactions of the same type submitted by a single IMS during the time period covered by the input IRUF.

A summarized IRUF can be used as input to

- the Performance Reporter and the Transaction Accountant
- additional TASCOSTR summarization runs
- IRUF utilities, programs PRSSELEC and PRSPRINT

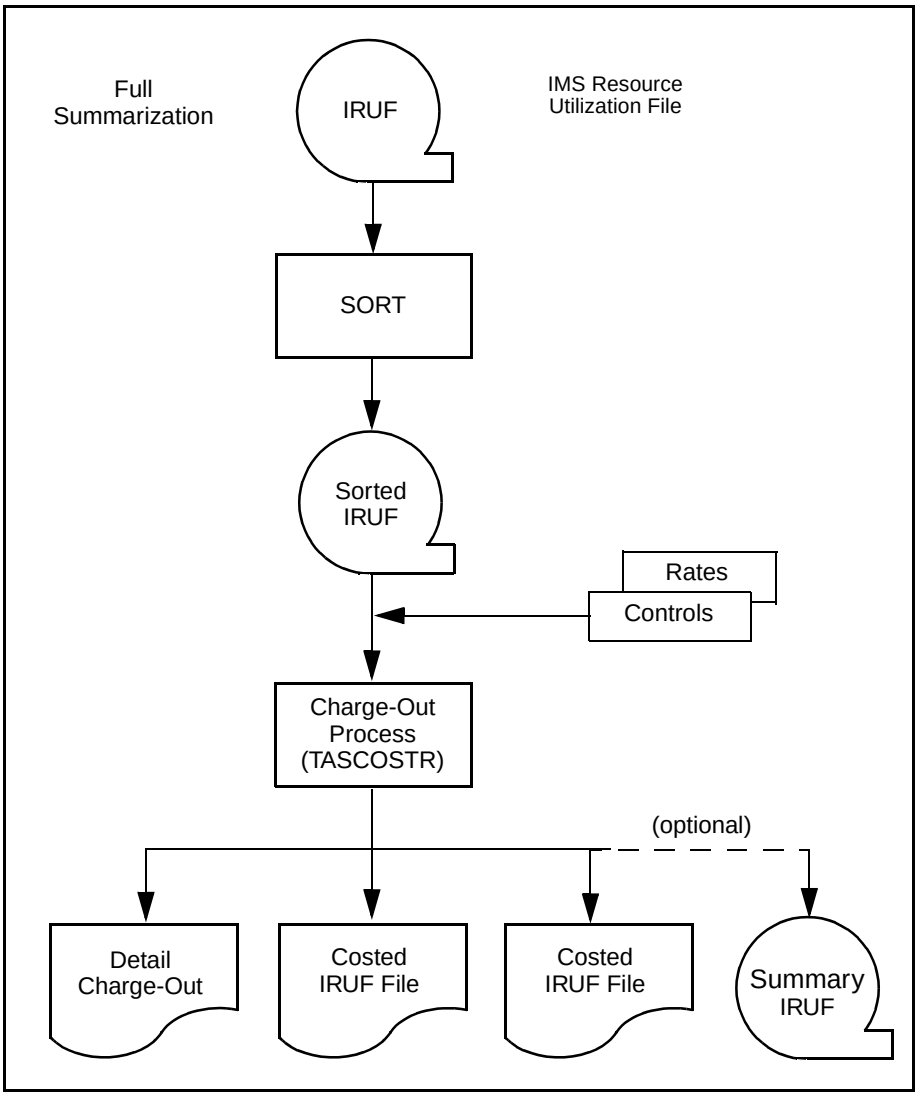
Summarization Process

IRUF summarization is performed by the TASCOSTR program. This program is the main driver of the MVIMS transaction accounting system. TASCOSTR generates a summarized IRUF, which is the standard output used in the accounting charge-out process. An illustration of the process is shown in Figure 12-1 on page 12-3.

You can also create a summarized IRUF without producing charge-out data. When you do so, TASCOSTR bypasses all accounting and charge-out processes provided by the program. For information about executing the TASCOSTR program for a specific offline component, see the *MAINVIEW for IMS Offline – Transaction Accountant Reference Manual* and the *MAINVIEW for IMS Offline – Performance Reporter Reference Manual*.

Tip: To create a summarized IRUF without generating reports, specify the SUMM option.

Figure 12-1 TASCOSTR Summarization Process



Summarization Considerations

The IRUF contains three record types, the transaction accounting record (TAR), the terminal (LTERM) accounting record (LAR), and the program accounting accounting record (PAR).

Transaction Accounting Records (TAR)

Transaction accounting records are summarized by transaction name within MVIMS customer ID. One record for a transaction per MVIMS customer ID quantifies all costs and resource usage on that transaction for the time period covered by the input IRUF or IRUFs.

In each summarized transaction accounting record, the database and terminal segments are summarized by unique identifiers. For each unique DBD name accessed by the transaction, there is a database segment. For each LTERM from which the transaction was submitted, there is a terminal segment.

Note: If the LTERM name is not part of the MVIMS customer ID, either physically or logically, the summarized transaction record for a user (identified by customer ID) can contain many terminal segments.

Customers who do not use the LTERM name as part of the MVIMS customer ID can include the CMPFMT32 parameter in the TASCOSTR EXEC statement to generate reports in the MVIMS version 3.2 format. If CMPFMT32 is specified, the MVIMS supported maximum is 50 segments, and if that maximum is exceeded, the summarization process stores data in an overflow segment called OTHERS. Some database names or LTERMs may be lost, but all the counts are preserved.

Terminal (LTERM) Accounting Records (LAR)

Terminal accounting records are summarized by LTERM name within MVIMS customer ID. One record for a terminal per MVIMS customer ID quantifies all resource usage for that terminal according to the time period covered by the input IRUF or IRUFs.

Program Accounting Records (PAR)

Program accounting records are summarized by program name. One record for a program name quantifies all resource usage by that program for the time period covered by the input IRUF or IRUFs.

Setting Fields and Summing Quantities

The summarization process is straightforward. Generally, the process is concerned only with setting characteristic fields and summing quantities. The following fields are automatically set and summarized:

Identification Fields

For a summarized IRUF, fields such as CLASS and MESSAGE REGION are set by the first value encountered in the summarization process.

Timestamps

In a detail IRUF, timestamps represent action times such as arrival time. In a summarized IRUF, timestamps represent time period bounds within which summarization is performed. However, the response option timestamps are those from the first detail record summarized, because bounds would be meaningless.

Resource Usage

Absolute Measures: In a detail IRUF, fields such as MEMORY AVAILABLE represent the actual use. In a summarized IRUF, these fields represent high-water marks.

Cumulative Measures: In a detail IRUF, fields such as CPU TIMES represent the CPU time used for an individual transaction. In a summarized IRUF, accumulations of all CPU time or counts used for the transactions within the summarization period are used.

Note: The response option fields for input queue time and response time will be accumulated if all transaction records that were summarized together were created with this option in effect; otherwise, the fields are set to zero.

User-Generated Response Time Segments

The response time fields in an IRUF are summarized as described in “Setting Fields and Summing Quantities” on page 12-5. The response time distributions produced by PRSRESP cannot show transaction level detail when based on summarized IRUF data.

The user has the option of specifying ranges for additional response data to be collected. When this option is used, the summarized transaction accounting record can contain one or more response segments. These segments are stored in the variable section of the record and are identified by an R in the segment-type code.

Note: These R-type segments are not processed by any MVIMS programs; however, they may be used as input to any user-written program.

Transaction response statistics are maintained in the response segment. Statistics are based on three criteria:

- actual response time values calculated for each transaction processed
- transaction category defined
- response time thresholds specified

Response Time Value

The response time is derived from the timestamps collected by the Event Collector. Normally, the response time is defined as the sum of INPUT QUEUE TIME + ELAPSED TIME. If the summarization process is performed with the response option in effect, the response time (TAR R RESPONSE) will be used in gathering these statistics. In this case, only transactions that made a response to the originating terminal will be counted.

Transaction Category Code

A major benefit of the transaction category code is that it allows the user to group transactions meaningfully. The user can establish this code through the PRSCEXIT exit routine. PRSCEXIT details are provided on page 8-5. The default value establishing the transaction category code is the transaction class used for processing that transaction.

Response Time Thresholds

Response time thresholds are parameters supplied by the user.
The parameters are used by TASCOSTR if the user chooses to create the response time segments in the summarized IRUF.

Parameters are based on transaction category, which indicate how response times should be accumulated for a summarized IRUF. For example, the user could specify that for all transactions in transaction category 006, statistics shall be maintained for the number of times the response time fell into each of the following response groups:

0 – 3	seconds
4 – 8	seconds
9 – 20	seconds
21 – 45	seconds
46 – 90	seconds
91 – n	seconds

With the preceding specifications, the summarization process constructs a response segment for each transaction record having a transaction category of 006. This process maintains statistics on how the transaction response time corresponded with the response time thresholds provided for the summarization time period.

Response Threshold Definition

The response thresholds provide a way to maintain statistics in a summarized IRUF that spans long periods of time. Response threshold definitions are submitted to the TASCOSTR program with the LISTRCNT DD statement, the same stream in which control level definitions are provided to the costing process (see the *MAINVIEW for IMS Offline – Performance Reporter Reference Manual* and the *MAINVIEW for IMS Offline – Transaction Accountant Reference Manual*). Table 12-1 on page 12-8 shows how to specify the response threshold statements.

Note: There are 11 response thresholds. The description assigned to threshold 02 is used for the remaining 9. Specify only the number of thresholds to be used, from lowest to highest, starting with threshold position 01. There are no defaults.

Table 12-1 Response Threshold Statements

Position	Data Element	Description
01-02	Identification	Value of RR expresses the response threshold in seconds. Value of RE expresses the response threshold in 1/100ths of a second. The maximum threshold is 99999 (expressed as 99999 seconds for RR and 999.99 seconds for RE). Both statements can be used together, but each one must specify a different category code.
03-05	Transaction category	Indicates the transaction category code for which the response thresholds in this statement are to be used.
06-10	Response threshold 01	Specifies a response group in which transactions will be counted if they have a response time greater than zero seconds but less than this threshold. The value is in seconds or 1/100ths of a second, depending on an identification of RR or RE.
11-15	Response threshold 02	Specifies a response group in which transactions will be counted if they have a response time greater than that specified in the previous threshold, but less than this threshold. The value is in seconds or 1/100ths of a second depending on an identification of RR or RE.
16-20	Response threshold 03	
21-25	Response threshold 04	
26-30	Response threshold 05	
31-35	Response threshold 06	
36-40	Response threshold 07	
41-45	Response threshold 08	
46-50	Response threshold 09	
51-55	Response threshold 10	
56-60	Response threshold 11	

Summarized IRUF Exit

TASCOSTR supports a user exit that allows you to change the IRUF file during creation. A sample COBOL exit is provided in BBSAMP member TASEXIT. This exit must be named TASEXIT and is link-edited into module TASCOSTR.

For more information about this user exit, see “Summarized IRUF Exit” on page 8-7.

Chapter 13 IRUF Utilities

MVIMS Offline provides two utilities for selecting and printing IRUF data, PRSSELEC and PRSPRINT. Both programs can process either detail or summarized IRUFs.

The PRSSELEC data selection utility selects records from an input IRUF. The utility can also print the input IRUF records or user-selected records in hexadecimal format. Users specify which record data is to be selected with PRSSELEC keyword statements. Selected records can be

- written to an extract file for further processing

  For example, the extract file could be used as input to the PRSPRINT program, which can print a formatted report of the record contents (shown on page 13-19).

- used to produce reports for a specified subset of the workload

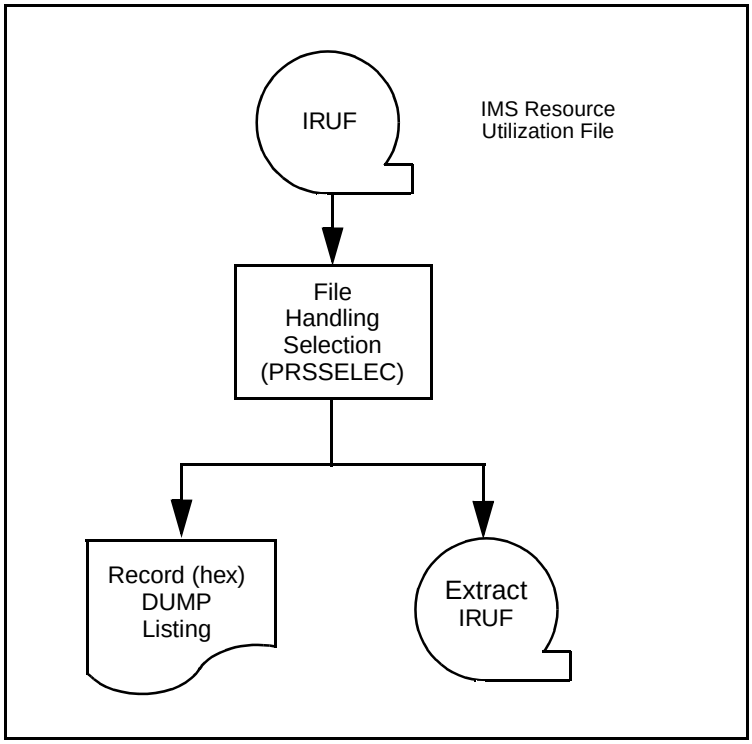
  For example, reports can be produced for all transactions associated with one user, and such reports can be useful for detailed analysis.

The PRSPRINT print utility reads every record of an input IRUF and prints a formatted MVIMS report showing the contents of the IRUF records. Since PRSPRINT does not have any selective printing capability, the utility is usually used with a data selection utility such as PRSSELEC, or with both a data selection utility and a sort utility.

Selecting Data from an IRUF (PRSSELEC)

The data selection utility PRSSELEC executes as a batch program. The utility's processing flow is shown in Figure 13-1.

Figure 13-1 PRSSELEC System Flow



The PRSSELEC program creates an IRUF extract file, prints all or portions of the IRUF, and selects user-specified records from the IRUF. Creating the IRUF extract file and printing the IRUF records are PRSSELEC execution options. IRUF record selections are made with PRSSELEC selection keyword statements.

Input and Output

IRUF data is the input to PRSSELEC. The IRUF input can be detail or summarized data. The IMFLEDIT program is executed against the IMS log to create detail IRUF data. The MVIMS TASCOSTR program is executed against a detail IRUF to create summarized IRUF data.

Output to PRSSELEC is controlled by input keyword selection statements and PARM execution options. Output can be an IRUF extract file or a hexadecimal IRUF record listing.

Note: PRSSELEC selects record occurrences. When a summarized IRUF is being processed, one record occurrence can represent many transactions. When an extract summary IRUF is processed, the reports produced can show a larger number of transactions than the number of records selected.

PRSSELEC Job Control Language Statements

To execute the PRSSELEC utility, use the JCL in Table 13-1. Figure 13-2 on page 13-4 provides an example of PRSSELEC JCL.

Table 13-1 PRSSELEC JCL Statements

Statement	Use
JOB	Initiates the job.
EXEC	Specifies the program name of the data handling utility (PRSSELEC), the region required (1024K), and the necessary parameter information (PARM=).
STEPLIB DD	Defines the program library (IMF.LOAD) that contains PRSSELEC program load module.
SELPRINT DD	Defines the message data set that is used to document syntax errors found in selection statements. The data set contains the Selection Function Diagnostic Messages report.
SYSPRINT DD	Defines the print output data set used to display the • records selected for printing • PRSSELEC Processing Diagnostics report, which contains parameter error and processing diagnostic messages • selection, print, and extract file output counts
INPUT DD	Defines the IMS Resource Utilization File (IRUF) used as input to the utility.
EXTRACT DD	Defines the file to contain records extracted from the input IRUF.
SYSIN DD	Defines the data set that contains the PRSSELEC selection keyword statements.

Figure 13-2 PRSSELEC JCL Example

```
//JOBNAME  JOB ..........
//STEP1    EXEC PGM=PRSSELEC,REGION=2048K,
//              PARM='PRTSEL,WRITE'
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SELPRINT DD   SYSOUT=A
//SYSPRINT DD   SYSOUT=A
//INPUT    DD   DSN=IRUF.MONTHS,DISP=SHR
//EXTRACT  DD   DSN=TEMP.IRUF,DISP=(NEW,KEEP),
//              UNIT=SYSDA,SPACE=(CYL,(4,1)),
//              DCB=(RECFM=VBS,LRECL=30970,
//              BLKSIZE=30974)
//SYSIN    DD   *
   SYSID=A
   TERMQUAL=PERS
   DBDNAME=PERSMAST
/*
```

In the sample JCL shown in Figure 13-2, the input is a monthly IRUF (IRUF.MONTHS). The utility execution options (PARM parameter in the EXEC statement) are to print selected records (PRTSEL) and to create an extract file of the selected records (WRITE). The records were created on System A (SYSID=A). The input is from a logical terminal name that has the first four characters of PERS (TERMQUAL=PERS). The database being accessed is PERSMAST (DBDNAME=PERSMAST).

PARM Options on the EXEC Statement

PRSSELEC provides three print options and two file creation options, which are specified with the PARM parameter in the JCL EXEC statement.

```
PARM='a,b'
```

The value `'a'` represents the following print file options:

PRTALL	Print all records read from the input IRUF (in hexadecimal format), and flag the selected records (all records if no selection is specified).
PRTSEL	Print only selected records from the input IRUF (in hexadecimal format).
NOPRNT	Do not invoke the print function.

The value `'b'` represents the following create file options:

NOWRT	Do not create an IRUF extract file of selected records.
WRITE	Create an IRUF extract file of selected records.

At the end of the job, PRSSELEC shows the number of records selected, printed, and written.

PRSSELEC Selection Options

PRSSELEC data selection options are specified with keyword parameters in data selection specification statements. Table 13-2 on page 13-6 shows the data selection options and their associated keyword parameters. Table 13-2 is provided for quick reference. For detailed descriptions of each keyword parameter, see “Selection Keyword Parameter Statements” on page 13-8.

PRSSELEC uses the following keys to select records from the IRUF.

Type Key	Data Type
T	Transaction accounting record data
P	Program accounting record data
L	Terminal accounting record data
D	Transaction accounting record, database segment data
E	Transaction accounting record, DB2 segment data
ALL	Transaction, program, and terminal accounting record data

Note: PRSSELEC applies the logical operator AND between keyword statements for each record type. It applies the logical operator OR to multiple parameters for one keyword.

Table 13-2 PRSSELEC Data Selection Options

User Option	Keyword Parameter	Data Type Key
Customer name	CUSTNAME=id	L|T
Customer name qualifier	CUSTQUAL=idsubset	L|T
Database name	DBDNAME=name	D
Database qualifier	DBDQUAL=namesub	D
DB2 application plan name	PLANNAME=name	E
DB2 application plan name qualifier	PLANQUAL=namesub	E
Fast Path routing code	ROUTNAME=code	T
Fast Path routing code qualifier	ROUTQUAL=codesub	T
Program name	PROGNAME=name	P|T
Program name qualifier	PROGQUAL=namesub	P|T
Program type	PGMTYPE=MPP, BMP, IFP, MDP, NDP, FPU, TPI, DBT, NOTDBT **Note:** The MPP program type includes JMP programs, the BMP type includes JBP programs, and DBT/NOTDBT includes CICS and ODBA threads.	P|T
Record number (sequential position of record within IRUF file)	NBR=number	ALL
Time period	PER=(yyddd,hhmm, yyddd,hhmm), SEC=(hhmm,hhmm), WITHIN=(TAR,PAR,LAR)	ALL
Record type	REC=L, P, T	ALL
Shared message queue group name	SMQGROUP=name	T
System ID	SYSID=id	ALL
Terminal name	TERMNAME=name	L|T
Terminal name qualifier	TERMQUAL=namesub	L|T
Transaction name	TRANNAME=name	T
Transaction name qualifier	TRANQUAL=namesub	T
Transaction type (DB2 subsystem or database access)	TRANTYPE=DLI, DLI-ONLY, DB2, DB2-ONLY, ALL, BOTH, NEITHER	T

Selection Statement Syntax

The following syntax rules apply to all the keyword selection statements:

- Selection statements cannot extend past column 71.
- Selection statements can begin in any column before column 71.
- Some selection statements are limited to one operand. Check individual selection statements for syntax rules.
- Multiple selection operands are separated by commas.
- Continuation is not allowed for any selection statement. If an operand does not fit in a particular statement, you must code another selection statement using the selection operator specified in the first one.
  For example:

```
TERMNAME=LLTER01,LLTER09,LLTER15,LLTER26,LLTER32
TERMNAME=LLTER05,LLTER11
```

- There is a limit to the number of criteria of a particular type that you can specify. Check individual selection statements for this limitation.
- Selection statements are validated for syntax and logically consistent field definitions (field size, numeric content, or numeric range). See the specification section of each criteria definition for exact requirements.
- Each statement is validity-checked until an error is encountered.
  No indication is provided of multiple errors in a single statement.

Note: The following pairs of keyword parameters have the logical operator OR applied if they are submitted in the same run of PRSSELEC:

TERMNAME and TERMQUAL
PROGNAME and PROGQUAL
CUSTNAME and CUSTQUAL
PLANNAME and PLANQUAL
TRANNAME and TRANQUAL
DBDNAME and DBDQUAL
ROUTNAME and ROUTQUAL

Selection Keyword Parameter Statements

This section describes the PRSSELEC IRUF data selection keyword parameters and lists them alphabetically by keyword.

You can specify keywords to selectively review specific IRUF record data. For example, you can review IRUF records generated during a specific time by specifying the PRSSELEC time period keyword statement.

Customer Name – CUSTNAME=

Selection:	CUSTNAME=id
Description:	Any valid 1- to 18-character MVIMS customer ID can be specified. MVIMS customer IDs are user-defined with a user exit from the Log Edit program IMFLEDIT.
Criteria:	N/A
Limit:	A maximum of 200 MVIMSs can be specified per run.
Example:	`CUSTNAME=CST01.S1001,CST02.S1001,SALES.FORECAST`

Customer Name Qualifier – CUSTQUAL=

Selection:	CUSTQUAL=idsubset
Description:	Any 1- to 17-character subset of any valid MVIMS customer ID can be specified. If specified, this keyword causes the selection of any MVIMS customer ID that begins with the characters specified in the operand. MVIMS customer IDs are user defined through a user exit from the Log Edit program IMFLEDIT.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`CUSTQUAL=CST01,CST02,SALES`

Database Name – DBDNAME=

Selection:	DBDNAME=name
Description:	Any valid 1- to 8-character database name can be specified.
Criteria:	N/A
Limit:	Up to 200 database names can be specified.
Example:	`DBDNAME=EMPLOYEE,PAYROLL9`

Database Name Qualifier – DBDQUAL=

Selection:	DBDQUAL=namesub
Description:	Any 1- to 7-character subset of any valid database name can be specified. If specified, this keyword causes the selection of any database name that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`DBDQUAL=PAY,PQ,Z`

Record Number – NBR=

Selection:	NBR=number
Description:	The number or range of numbers (sequentially from the beginning) that indicate the position of a particular record or group of records in the input IRUF. A number range is indicated with a hyphen. A numeric check is performed on each field.
Criteria:	Any valid numeric up to 99999 is allowed.
Limit:	Up to 40 record numbers can be specified.
Example:	`NBR=1,2,83,17-51,91`

Time Period – PER=,SEC=,WITHIN=

Selection:	PER=(period),SEC=(section),WITHIN=(TAR,PAR,LAR)
Description:	A time period (PER) is specified as a range of dates and times, low to high, for the period to be selected. Also, within the time period, a time section (SEC) can be specified. For example, if the time 0800 to 1600 was selected for the month of July, the time period would cover the first to last day of July including the time to start and stop the selection on those two days. The time selection represents an additional qualification for each day of July. **Tip:** Both PER and SEC select all records where any time data overlaps with the selection criteria. To reduce the selection window to records that contain only time data that is completely within the times requested, you must add the WITHIN parameter. Because long times are frequently associated with terminal (LAR) records, MVIMS provides an option to limit the effect of the WITHIN parameter. In most cases specifying WITHIN=(TAR,PAR) suffices. **Note:** A time of 2359 is changed internally to 23:59:59:99 to encompass data timestamped between 23:59 and 00:00. Midnight is specified as 0000 (rather than as 2400). PER= Specified from left to right as low date and time to high date and time in the format (yyddd,hhmm,yyddd,hhmm). The value must be enclosed in parentheses, and individual operands are separated with commas. yy last two digits of the year ddd Julian day (1–366) hh hour (00–23) mm minute (00–59) SEC= Specified from left to right as low date and time to high date and time in the format (hhmm,hhmm). The value must be enclosed in parentheses, and individual operands are separated with commas. hh hour (00–23) mm minute (00–59) WITHIN= The additional restriction on the selection of the record is in effect only for the specified IRUF record type.
Criteria:	Any valid time period and time section can be specified. **Note:** For a summarized IRUF, if you select a shorter time period than the summarization period, all records with dates that span the selection period will be selected. Reports from the extract file still show the IRUF summarization period. For example, if one day is selected from a weekly IRUF, reports would still contain weekly information.
Limit:	One time period and section can be specified per statement and there can be up to a maximum of three PER statements. Time period selections do not have to be in input stream order.
Example:	`PER=(85182,001,85212,0000),SEC=((0800,1000), ... X (1200,1400),(1600,1800)),WITHIN=(TAR,PAR,LAR)` The X continuation indicator must be in column 80.

Program Type – PGMTYPE=

Selection:	PGMTYPE=type
Description:	Type can be MPP Message processing program BMP Batch message processing program MDP Fast Path message-driven program NDP Fast Path non-message-driven program FPU Fast Path utility IFP IMS Fast Path program TPI CPI-C driven program DBT DBCTL threads NOTDBT All threads excluding DBCTL threads
Criteria:	MPP, BMP, MDP, NDP, FPU, IFP, TPI, DBT, NOTDBT
Limit:	Up to the maximum allowed. More than one type can be specified.
Example:	`PGMTYPE=MPP,BMP,MDP`
Note:	The MPP program type includes JMP programs, the BMP type includes JBP programs, and DBT/NOTDBT includes both CICS and ODBA threads.

Plan Name – PLANNAME=

Selection:	PLANNAME=name
Description:	Any valid 1- to 8-character DB2 application plan name can be specified.
Criteria:	N/A
Limit:	Up to 200 plan names can be specified.
Example:	`PLANNAME=BS100,BS210`

Plan Name Qualifier – PLANQUAL=

Selection:	PLANQUAL=namesub
Description:	Any 1- to 7-character subset of any valid DB2 application plan name can be specified. If specified, this keyword causes the selection of any application plan name that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`PLANQUAL=BS1,ZS,Z`

Program Name – PROGNAME=

Selection:	PROGNAME=name
Description:	Any valid 1- to 8-character program name can be specified.
Criteria:	N/A
Limit:	Up to 200 program names can be specified.
Example:	`PROGNAME=PSB63000,FAS01888`

Program Name Qualifier – PROGQUAL=

Selection:	PROGQUAL=namesub
Description:	Any 1- to 7-character subset of any valid program name can be specified. If specified, this keyword causes the selection of any program name that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`PROGQUAL=PSB6,FAS,DD`

Record Type – REC=

Selection:	REC=type
Description:	A 1-character field that identifies the IRUF record type, which can be L Terminal (LTERM) accounting record (LAR) P Program accounting record (PAR) T Transaction accounting record (TAR)
Criteria:	Selection is made against the three record types contained in an IRUF.
Limit:	One to three record types can be specified.
Example:	`REC=L,T`

Fast Path Routing Code – ROUTNAME=

Selection:	ROUTNAME=code
Description:	Any valid 1- to 8-character routing code assigned to a particular Fast Path transaction can be specified.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`ROUTNAME=TPAY0005,TRZZ9083`

Fast Path Routing Code Qualifier – ROUTQUAL=

Selection:	ROUTQUAL=codesub
Description:	Any 1- to 7-character subset of any valid Fast Path routing code can be specified. If specified, this keyword causes the selection of any routing code that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`ROUTQUAL=PAY,REC`

SMQ (Shared Message Queue) Group Name – SMQGROUP=

Selection:	SMQGROUP=name
Description:	Any valid 1- to 8-character SMQ group name can be specified.
Criteria:	N/A
Limit:	Up to 200 SMQ group names can be specified.
Example:	`SMQGROUP=IMSSMQ1,IMSSMQ2`

System Identifier – SYSID=

Selection:	SYSID=id
Description:	Any valid user-assigned MVIMS system identifier can be specified. Each IRUF record has a SYSID field.
Criteria:	N/A
Limit:	Up to 20 system identifiers can be specified.
Example:	`SYSID=A,B,1,C,Z`

Terminal Name – TERMNAME=

Selection:	TERMNAME=name
Description:	Any valid 1- to 8-character LTERM name can be specified.
Criteria:	N/A
Limit:	Up to 200 terminal names can be specified.
Example:	`TERMNAME=LTERM4,LMHPAYRL,PHEALTH`

Terminal Name Qualifier – TERMQUAL=

Selection:	TERMQUAL=namesub
Description:	Any 1- to 7-character subset of any valid LTERM name can be specified. If specified, this keyword causes the selection of any LTERM name that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`TERMQUAL=11,PH,LTERM`

Transaction Name – TRANNAME=

Selection:	TRANNAME=name
Description:	Any valid 1- to 8-character transaction name assigned to a particular transaction can be specified.
Criteria:	N/A
Limit:	Up to 200 transaction names can be specified.
Example:	`TRANNAME=TPAY0005,TRZZ9083`

Transaction Name Qualifier – TRANQUAL=

Selection:	TRANQUAL=namesub
Description:	Any 1- to 7-character subset of any valid transaction name can be specified. If specified, this keyword causes the selection of any transaction name that begins with the characters specified in the operand.
Criteria:	N/A
Limit:	Up to 200 qualifiers can be specified.
Example:	`TRANQUAL=PAY,II7`

Transaction Type – TRANTYPE=

Selection	TRANTYPE=type
Description:	Type can be DLI — Transactions that access DL/I databases DLI-ONLY — Transactions that access only DL/I databases DB2 — Transactions that access the DB2 subsystem DB2-ONLY — Transactions that access only the DB2 subsystem ALL (default) — All transactions BOTH — Only transactions that access both the DB2 subsystem and DL/I databases NEITHER — Transactions that have no DL/I or DB2 activity
Criteria:	DLI, DLI-ONLY, DB2, DB2-ONLY, ALL, BOTH, or NEITHER
Limit:	Only one type can be specified.
Example:	`TRANTYPE=DB2`

PRSSELEC Diagnostic Reports

This section provides examples of the diagnostic reports produced by the PRSSELEC utility. PRSSELEC produces the following two diagnostic reports:

- Selection Function Diagnostic Messages report
- Processing Diagnostics report

Selection Function Diagnostic Messages Report

The Selection Function Diagnostic Messages report contains PRSSELEC statement syntax messages. A sample of the report is shown in Figure 13-3.

Figure 13-3 PRSSELEC Selection Function Diagnostic Messages Report

```
*** IMF ****                                    IMF  UTILITIES                                    **** IMF ****
                                        SELECTION FUNCTION DIAGNOSTIC MESSAGES
REC=T
****IMF004  END INPUT -- NO ERRORS DETECTED
```

Processing Diagnostics Report

The Processing Diagnostics report contains PRSSELEC parameter and processing diagnostic messages. A sample of the report is shown in Figure 13-4.

Figure 13-4 PRSSELEC Processing Diagnostics Report

```
**** IMF ****                                    IMF UTILITIES                                     **** IMF ****
                                        PRSSELEC PROCESSING DIAGNOSTICS
**** PARAMETER OPTIONS FOR THIS RUN ARE    'NOPRNT,WRITE'

END OF INPUT -- SUCCESSFUL TERMINATION
  RECORDS SELECTED      042133
  RECORDS PRINTED       000000
  RECORDS WRITTEN       042133
     BREAKDOWN:   TAR WRITTEN      042133
                  PAR WRITTEN      000000
                  LAR WRITTEN      000000
```

Printing an IRUF (PRSPRINT)

This section describes the MVIMS Offline print utility PRSPRINT, a batch program that produces a formatted report of the records read from an input IRUF.

PRSPRINT does not provide data selection. The data selection utility PRSSELEC or a data selection and sort utility must be run against the IRUF to produce an input extract IRUF for selective PRSPRINT reporting.

Input and Output

Input to PRSPRINT can be either of the following:

- IRUF data produced by IMFLEDIT (detail IRUF) or TASCOSTR (summarized IRUF)
- Extracted IRUF data created by PRSSELEC

A sample of a PRSPRINT-formatted IRUF report is shown in Figure 13-6 on page 13-19.

PRSPRINT Job Control Language Statements

PRSPRINT is a one-step process. Use the JCL statements in Table 13-3 to execute the procedure (Figure 13-5 on page 13-18 provides an example of PRSPRINT JCL).

Table 13-3 PRSPRINT JCL Statements

Statement	Use
JOB	Initiates the job.
EXEC	Specifies the program name of the data handling utility (PRSPRINT) and the size of the region required to run the program.
STEPLIB DD	Defines the program library (IMF.LOAD) that contains the PRSPRINT program load module.
SYSPRINT DD	Defines the print output data set used to display the formatted IRUF report. The DCB attributes of the data set are `RECFM=FB,LRECL=133,BLKSIZE=133`
RESUTIL DD	Defines an input IRUF. The DCB attributes of the data set are `DCB=(RECFM=VBS,LRECL=30970,BLKSIZE=30974)`

Figure 13-5 PRSPRINT JCL Example

```
//JOBNAME  JOB  ..........
//STEP1    EXEC PGM=PRSPRINT,REGION=1024K
//STEPLIB  DD   DSN=IMF.LOAD,DISP=SHR
//SYSPRINT DD   SYSOUT=A
//RESUTIL  DD   DSN=IRUF.MONTHS,DISP=SHR
//
```

Formatted IRUF Record Report

The PRSPRINT program produces a specially formatted report that shows the contents of the IRUF records. The report is called the Formatted IRUF Record Report, and an example of the report is provided in Figure 13-6 on page 13-19.

An IRUF data set can contain three types of records:

- transaction accounting records (TAR)
- program accounting records (PAR)
- terminal (LTERM) accounting records (LAR)

Transaction records can contain a variable number of fixed-length trailers or segments.

When processing the IRUF records, PRSPRINT reads and prints every record contained in the input IRUF defined by the RESUTIL DD statement. PRSPRINT prints a record header, a fixed-length data area, and any trailers that are present. Records and trailers are separated by a line of hyphens.

Data elements for each of the three IRUF record types (TAR, PAR, and LAR) are described in Table 13-4 on page 13-21, Table 13-5 on page 13-27, and Table 13-6 on page 13-29.

Figure 13-6 Formatted IRUF Record Report

```
**** IMF ****                                    IMF UTILITIES                                                    **** IMF ****
CURRENT DATE - 03/22/yy   TIME - 1053          FORMATTED IRUF RECORD REPORT                                        PAGE -    1
*****************************************************************************************************************************
* RECTYPE-T TIME-05:42:19.12   DATE-2002066                                                                                  *
*                              * SUBTYPE = D          * VERSION = 01         * SYSID = 1            * IMS-LEVL = xx00        *
*                              * TRANCODE= KNV9Z3BS   * PSBNAME = KNV9Z41    * CUSTID1 = K0771190   * CUSTID2  =              *
*                              * LTERM   = K0771190   * REGION  = A17MR100   * CLASS   = 001        * TRANTYPE =              *
*                              * TRANCAT = 001        * USER-ID = K0771190   * AGN     =            * OS-CODE  = MVS         *
*                              * #-TRANS = 000000001  * R-OPTION= R          * A-DATE  = yy152       * A-TIME   = 05:42:19.00 *
*                              * S-DATE  = yy152      * S-TIME  = 05:42:19.12 * E-DATE  = yy152       * E-TIME   = 05:42:19.23 *
*                              * I-Q-TIME= 00:00:00.12 * ELAPSED = 00:00:00.11 * MSG-CPU = 00:00:00.012* OPEN-CPU = 00:00:00.000*
*                              * MSG-DLI = 00:00:00.000* CTL-DLI = 00:00:00.001* MSG-BUF = 00:00:00.000* CTL-BUF  = 00:00:00.000*
*                              * DB2ID   =            * DB2-CPU = 00:00:00.000* DB2-TRAN= 000000000   *                        *
*                              * A-CORE  = 0952       * U-CORE  = 0020        * A-KCOREM= 04294965.551* U-KCOREM = 00000000.037*
*                              * R-MSWCNT= 0000       * R-MSNTCT= 0001        * R-MDQCNT= 0001        * R-MSWLEVL= 0000        *
*                              * R-ORIGDT= yy152      * R-ORIGTM= 05:42:19.00 * R-ADATE = yy152       * R-ATIME  = 05:42:19.00 *
*                              * R-IDATE = yy152      * R-ITIME = 05:42:19.20 * R-SDATE = yy152       * R-STIME  = 05:42:19.20 *
*                              * R-DDATE = yy152      * R-D-TIME= 05:42:24.10 * R-LINE  = 0000        * R-PTERM  = 0000        *
*                              * R-INPUTQ= 00:00:00.12 * R-OUTPTQ= 00:00:00.00 * R-DE-Q  = 00:00:04.90 * R-RESPON = 00:00:00.20 *
*                              * $-ACORE = $000000.000 * $-UCORE = $000000.000 * $MSGCPU = $000000.000 * $-MIN    = $000000.000 *
*                              * $-MSGDLI= $000000.000 * $-CTLDLI= $000000.000 * $-RESRCE= $000000.000 * $-UNIT   = $000000.000 *
*                              * $-PRORAT= $000000.000 * $-BILABL= $000000.000 * $-VAR   = $000000.000 * $DB2CPU  = $000000.000 *
*                              * SAP FLG = N          * U-AREA1 =            * U-AREA2 =            * #-SEGS   = 002         *
*                              * SPRC    = 0000       * BGQCT   = 0000        * MSG-OPN = 00:00:00.000* RTCODE   =             *
*                              * NBA     = 0000       * OBA     = 0000        * BUSED   = 0000        *                        *
*                              * NCIC    = 0000       * NWFB    = 0000        * WFOBL   =             *                        *
*                              ------------------------------------------------------------------------------------------------ *
*                              * SEGTYPE = P          * LTERM   = K0771190    * LINENBR = 000000000   * CON-TIME = 00:00:00.00 *
*                              * #-MGU   = 000000001  * #-MGN   = 000000000   * #-MISRT = 000000001   * #-MPURGE = 000000000   *
*                              * #-MOTHR = 000000000  * CNT-IN  = 000000000114* IN-SPA  = 000000000000* IN-OTHR  = 000000000000*
*                              * CNT-OUT = 000000000145* OUT-SPA = 000000000000* OUT-ATER= 000000000000* OUT-ATRN = 000000000000*
*                              * OUT-OTHR= 000000000000*                      *                       *                        *
*                              * $-CONECT= $000000.000 * $-MGU   = $000000.000 * $-MGN   = $000000.000 * $-MISRT  = $000000.000 *
*                              * $-MPURGE= $000000.000 * $-MOTHR = $000000.000 * $-INPUT = $000000.000 * $-IN-SPA = $000000.000 *
*                              * $-IN-OTH= $000000.000 * $-OUTPUT= $000000.000 * $-OUTSPA= $000000.000 * $-OUTATER= $000000.000 *
*                              * $-O-ATRN= $000000.000 * $-OTHER = $000000.000 *                       *                        *
*                              ------------------------------------------------------------------------------------------------ *
*                              * SEGTYPE = D          * DBDNAME = KZZP86      * DBDORG  = 1           * DMB-SIZE = 000000416   *
*                              * #-GU    = 000000001  * #-GN    = 000000000   * #-ISRT  = 000000000   * #-DELETE = 000000000   *
*                              * #-REPL  = 000000000  * #-OTHER = 000000000   * #-ISAMR = 000000000   * #-ISAMW  = 000000000   *
*                              * #-OSAMR = 000000002  * #-OSAMW = 000000000   * #-REFS  = 000000001   * #-DB2DLI = 000000000   *
*                              * #-NO-I  = 000000001  * #-NO-O  = 000000000   *                       *                        *
*                              * $-GU    = $000000.000 * $-GN    = $000000.000 * $-ISRT  = $000000.000 * $-DELETE = $000000.000 *
*                              * $-REPL  = $000000.000 * $-OTHER = $000000.000 * $-ISAMR = $000000.000 * $-ISAMW  = $000000.000 *
*                              * $-OSAMR = $000000.000 * $-OSAMW = $000000.000 * $-NO-I  = $000000.000 * $-NO-O   = $000000.000 *
*                              ------------------------------------------------------------------------------------------------ *
* RECTYPE-P  TIME-05:42:37.90  DATE-2002066                                                                                  *
*                              * SUBTYPE = D          * VERSION = 01         * SYSID   = 1           * IMS-LEVL = xx00        *
*                              * PSBNAME = KUM9Z80    * REGION  = A17MR101    * PSBTYPE =            * OS-CODE  = MVS         *
*                              * S-DATE  = yy152      * S-TIME  = 05:42:37.90 * E-DATE  = yy152       * E-TIME   = 05:42:38.26 *
*                              * ELAPSED = 00:00:00.36 * SKED-CPU= 00:00:00.010* OVHD-CPU= 00:00:00.020*                       *
*                              * #-PROGS = 000000001  * S-ABENDS= 0000        * U-ABENDS= 0000        * AB-CODE  =             *
*                              * #-MGU   = 000000002  * #-MGN   = 000000000   * #-MISRT = 000000001   * #-MPURGE = 000000000   *
*                              * #-MOTHR = 000000000  * A-CORE  = 0952        * U-CORE  = 0018        * A-KCOREM = 04294961.585*
*                              * PSB-SIZE= 000049196  * DMB-SIZE= 000036448   * IMS-CPU = 00:00:00.040* U-KCOREM = 00000000.108*
*                              * IMS-SD  = yy152      * IMS-ST  = 05:42:37.90 * IMS-ED  = yy152       * IMS-ET   = 05:42:38.26 *
*                              * SAP FLG = N          *         =             *         =             *         =              *
*                              * U-AREA1 =            * U-AREA2 =             * AGN     =             * DB2ID    =              *
*                              * NBA     = 0000       * OBA     = 0000        * BUHWM   = 0000        * DB2-TRAN = 000000000   *
*                              ------------------------------------------------------------------------------------------------ *
*                                                                                                                            *
```

Record Header

Every record printed by PRSPRINT is prefaced by a record header. The header identifies the type of record encountered and the date/time of the activity.

Field Label	Definition	
RECTYPE	The type of record printed:	
	T	transaction accounting record
	P	program accounting record
	L	terminal accounting record
TIME/DATE	The time and date when the activity occurred. For the following record types, this value represents	
	T	time when the transaction processing started
	P	time when scheduling began for this PSB
	L	time when the terminal session started

Report Fields

The fields printed from a specific fixed-length data area or from a trailer are obtained from the IRUF record. To help you relate the fields printed in the IRUF report to the IRUF record fields, field mapping tables are included in the following sections for each fixed-length data area and segment type. For information about the record format of each segment type (SEGTYPE report field), see Appendix B, "IRUF Record Layout Descriptions."

Transaction Accounting Record (TAR) Data Elements for the PRSPRINT Report

Table 13-4 shows the TAR data elements for the labels in the IRUF report created by the PRSPRINT utility. For more information about the TAR record, see Appendix B, "IRUF Record Layout Descriptions."

Table 13-4 TAR Data Elements for PRSPRINT (Part 1 of 6)

Label	Data Element Name	Definition
SUBTYPE	TAR-RECORD-TYPE	Level of data in this record
VERSION	TAR-RECORD-VERSION	IRUF release identifier
SYSID	TAR-SYSTEM-ID	Originating system
IMS-LEVL	TAR-IMS-LEVEL	IMS release level
TRANCODE	TAR-TRANSACTION-CODE	Code identifying this message
PSBNAME	TAR-PROGRAM-NAME	Program name
CUSTID1	TAR-CUSTOMER-ID	Customer ID (first 9 bytes)
CUSTID2	TAR-CUSTOMER-ID	Customer ID (last 9 bytes)
LTERM	TAR-LOG-TERMINAL-NAME	Logical terminal
REGION	TAR-REGION-ID	Processing region name
CLASS	TAR-INPUT-CLASS	Transaction input class
TRANTYPE	TAR-TRANS-TYPE	Transaction processing type
TRANCAT	TAR-DET-TRANS-CATEGORY	Transaction category
USER-ID	TAR-USER-ID	Security system user ID
AGN	TAR-AGN	Application group number
OS-CODE	TAR-OS-CODE	Operating system code
#-TRANS	TAR-NO-TRANSACTIONS	Number of transactions run
R-OPTION	TAR-R-OPTION	Response option flag
A-DATE	TAR-TRANS-ARRIVAL-DATE	Arrival date
A-TIME	TAR-TRANS-ARRIVAL-TIME	Arrival time
S-DATE	TAR-TRANS-START-DATE	Processing start date
S-TIME	TAR-TRANS-START-TIME	Processing start time
E-DATE	TAR-TRANS-STOP-DATE	Processing end date
E-TIME	TAR-TRANS-STOP-TIME	Processing end time
I-Q-TIME	TAR-INPUT-Q-TIME	Time spent in queue
ELAPSED	TAR-ELAPSED-TIME	Transaction elapsed time
MSG-CPU	TAR-MSG-PROG-CPU-TIME	Message region CPU time
OPEN-PU	TAR-CTL-OPNCLS-CPU-TIME	Open/close CPU time
MSG-DLI	TAR-MSG-DL1-CPU-TIME	Message region DLI CPU time
CTL-DLI	TAR-CTL-DL1-CPU-TIME	Control region DLI CPU time

Table 13-4 TAR Data Elements for PRSPRINT (Part 2 of 6)

Label	Data Element Name	Definition
MSG-BUF	TAR-MSG-BUFFER-CPU-TIME	Message region buffer CPU time
CTL-BUF	TAR-CTL-BUFFER-CPU-TIME	Control region buffer CPU time
DB2ID	TAR-DB2-SUBSYS-ID	DB2 subsystem ID
DB2-CPU	TAR-DEP-DB2-CPU-TIME	Dependent region CPU time to process DB2 requests
DB2-TRAN	TAR-DB2-NO-TRANS	Number of transactions that accessed DB2
A-CORE	TAR-CORE-ALLOCATED	Allocated core
U-CORE	TAR-CORE-USED	Used core
A-KCOREM	TAR-ALLOC-KCORE-MINS	Allocated K-core minutes
U-KCOREM	TAR-USED-KCORE-MINS	Used K-core-minutes
Note: If the IRUF (with response option) is created, the following R- data elements (up to the next double line) are also printed from the fixed portion of the transaction record.		
R-MSWCNT	TAR-R-MSGSW-COUNT	Message switch count
R-MSNTCT	TAR-R-MSGSENT-COUNT	Message sent count
R-MDQCNT	TAR-R-MSGDEQ-COUNT	Message dequeue count
R-MSWLEVL	TAR-R-MSGSW-LEVEL	Message switching Level
R-ORIGDT	TAR-R-ORIGINAL-DATE	Original arrival date
R-ORIGTM	TAR-R-ORIGINAL-TIME	Original arrival time
R-ADATE	TAR-R-ARRIVAL-DATE	Message switch arrival date
R-ATIME	TAR-R-ARRIVAL-TIME	Message switch arrival time
R-IDATE	TAR-R-MSGISRT-DATE	Output message insert date
R-ITIME	TAR-R-MSGISRT-TIME	Output message insert time
R-SDATE	TAR-R-MSGSENT-DATE	Output message sent date
R-STIME	TAR-R-MSGSENT-TIME	Output message sent time
R-DDATE	TAR-R-MSGDEQ-DATE	Output message dequeue date
R-D-TIME	TAR-R-MSGDEQ-TIME	Output message dequeue time
R-LINE	TAR-R-INPUT-LINE	Input line number
R-PTERM	TAR-R-INPUT-PTERM	Input physical terminal
R-INPUTQ	TAR-R-INPUT-Q	Alternate input queue time
R-OUTPTQ	TAR-R-OUTPUT-Q	Alternate output queue time
R-DE-Q	TAR-R-DE-Q	Alternate dequeue time
R-RESPON	TAR-R-RESPONSE	Alternate response time
$-ACORE	TAR-CORE-ALOC-CHG	Charge for allocated K-core
$-UCORE	TAR-CORE-USED-CHG	Charge for used K-core
$-MSGCPU	TAR-MSG-PROG-CPU-CHG	Charge for message region CPU time
$-MIN	TAR-MINIMUM-CHARGE	Minimum transaction charge
$-MSGDLI	TAR-MSG-DL1-CPU-CHG	Charge for message region DLI CPU time

Table 13-4 TAR Data Elements for PRSPRINT (Part 3 of 6)

Label	Data Element Name	Definition
$-CTLDLI	TAR-CTL-DL1-CPU-CHG	Charge for control region DLI CPU time
$-RESRCE	TAR-RESOURCE-CHARGE	Total resource charges
$-UNIT	TAR-UNIT-CHARGE	Transaction unit charge
$-PRORAT	TAR-PRORATE-CHARGE	Total prorated charges
$-BILABL	TAR-BILLABLE-CHARGE	Total charge
$-VAR	TAR-CHARGE-VARIANCE	Charge variance
$DB2CPU	TAR-DEP-DB2-CPU-TIME	Charge for dependent region DB2 CPU time
SAP FLG	TAR-SAP-FLAG	SAP exit-invoked indicator (Y or N)
U-AREA1	TAR-USER-AREA	User area (first 10 bytes)
U-AREA2	TAR-USER-AREA	User area (last 10 Bytes)
#-SEGS	TAR-NO-DBD-SEGMENTS	Number of segments
SPRC	TAR-SPRC	Fast Path sync point return code
BGQCT	TAR-BGQCT	Fast Path balancing group queue count
MSG-OPN	TAR-MSG-OPNCLS-CPU-TIME	Dependent region open/close CPU time
RTCODE	TAR-ROUTING-CODE	Fast Path routing code
NBA	TAR-NBA	Fast Path normal buffer allocation
OBA	TAR-OBA	Fast Path overflow buffer allocation
BUSED	TAR-BUSED	Fast Path buffers used
NCIC	TAR-NCIC	Number of Fast Path CI contentions
NWFB	TAR-NWFB	Number of waits for Fast Path buffers
WFOBL	TAR-FLAG2	Wait for Fast Path overflow buffer latch
Note: The following data elements (up to the next double line) are in the trailer segment types of the transaction record for **terminal I/O data**.		
SEGTYPE	TAR-SEGMENT-PCB-TYPE	I/O PCB segment
LTERM	TAR-TERMINAL-NAME	Originating LTERM
LINENBR	TAR-PCB-LINENBR	PCB line number
CON-TIME	TAR-CONNECT-TIME	Connect time
#-MGU	TAR-MSG-GU-COUNT	Number of message get uniques
#-MGN	TAR-MSG-GN-COUNT	Number of message get nexts
#-MISRT	TAR-MSG-IN-COUNT	Number of message inserts
#-MPURGE	TAR-MSG-PU-COUNT	Number of message purges
#-MOTHR	TAR-MSG-OT-COUNT	Number of other message calls
CNT-IN	TAR-INPUT-CHAR-COUNT	Number of bytes input
IN-SPA	TAR-INPUT-SPA-COUNT	Number of bytes input to SPA
IN-OTHR	TAR-INPUT-OTHER-COUNT	Number of other bytes input
CNT-OUT	TAR-OUTPUT-CHAR-COUNT	Number of bytes output

Table 13-4 TAR Data Elements for PRSPRINT (Part 4 of 6)

Label	Data Element Name	Definition
OUT-SPA	TAR-OUTPUT-SPA-COUNT	Number of bytes output from SPA
OUT-ATER	TAR-OUTPUT-ALTTERM-COUNT	Number of bytes output from alternate area
OUT-ATRN	TAR-OUTPUT-ALTTRAN-COUNT	Number of bytes output from alternate message
OUT-OTHR	TAR-OUTPUT-OTHER-COUNT	Number of other bytes output
$-CONECT	TAR-CONNECT-CHG	Charge for connect time
$-MGU	TAR-MSG-GU-CHG	Charge for message get uniques
$-MGN	TAR-MSG-GN-CHG	Charge for message get nexts
$-MISRT	TAR-MSG-IN-CHG	Charge for message inserts
$-MPURGE	TAR-MSG-PU-CHG	Charge for message purges
$-MOTHR	TAR-MSG-OT-CHG	Charge for other characters
$-INPUT	TAR-INPUT-CHAR-CHG	Charge for input characters
$-IN-SPA	TAR-INPUT-SPA-CHG	Charge for input space characters
$-IN-OTH	TAR-INPUT-OTHER-CHG	Charge for other input characters
$-OUTPUT	TAR-OUTPUT-CHAR-CHG	Charge for output characters
$-OUTSPA	TAR-OUTPUT-SPA-CHG	Charge for output space characters
$-OUTATER	TAR-OUTPUT-ALTTERM-CHG	Charge for alternate characters
$-O-ATRN	TAR-OUTPUT-ALTTRAN-CHG	Charge for alternate message queue characters
$-OTHER	TAR-OUTPUT-OTHER-CHG	Charge for other characters
Note: The following data elements (up to the next double line) are in the trailer segment types of the transaction record quantifying **DL/I database utilization**.		
SEGTYPE	TAR-SEGMENT-DBD-TYPE	DBD segment
DBDNAME	TAR-DBD-NAME	DBD name
DBDORG	TAR-ORG-CODE	DBD organization
DMB-SIZE	TAR-DMB-POOL-SIZE	DMB pool requirements
#-GU	TAR-GU-COUNT	Number of get uniques
#-GN	TAR-GN-COUNT	Number of get nexts
#-ISRT	TAR-IN-COUNT	Number of inserts
#-DELETE	TAR-DL-COUNT	Number of deletes
#-REPL	TAR-RP-COUNT	Number of replaces
#-OTHER	TAR-OTHER-COUNT	Number of other DLI calls
#-ISAMR	TAR-ISAM-READS	Number of key reads
#-ISAMW	TAR-ISAM-WRITES	Number of key writes
#-OSAMR	TAR-OSAM-READS	Number of nonkey reads
#-OSAMW	TAR-OSAM-WRITES	Number of nonkey writes
#-REFS	TAR-NO-REFERENCES	Number of transactions that referenced this database
#-DB2DLI	TAR-DB2-DLI-CNT	Number of DLI calls made if a transaction also accesses DB2

Table 13-4 TAR Data Elements for PRSPRINT (Part 5 of 6)

Label	Data Element Name	Definition
#-NO-I	TAR-NO-I	Number of no inputs
#-NO-O	TAR-NO-O	Number of no outputs
$-GU	TAR-GU-CHG	Charge for get uniques
$-GN	TAR-GN-CHG	Charge for get nexts
$-ISRT	TAR-IN-CHG	Charge for inserts
$-DELETE	TAR-DL-CHG	Charge for deletes
$-REPL	TAR-RP-CHARGE	Charge for replaces
$-OTHER	TAR-OTHER-CHG	Charge for other DLI calls
$-ISAMR	TAR-ISAM-READ-CHG	Charge for key reads
$-ISAMW	TAR-ISAM-WRITE-CHG	Charge for key writes
$-OSAMR	TAR-OSAM-READ-CHG	Charge for nonkey reads
$-OSAMW	TAR-OSAM-WRITE-CHG	Charge for nonkey writes
$-NO-I	TAR-NO-I-CHG	Charge for no inputs
$-NO-O	TAR-NO-O-CHG	Charge for no outputs
Note: The following data elements (up to the next double line) are in the trailer segment types of the transaction record quantifying **DB2 database utilization**.		
PLANNAME	TAR-DB2-PLAN-NAME	DB2 plan name for transaction
#-SEL/F	TAR-DB2-SEL-FETCH-COUNT	Number of SQL select and fetch calls
#-OPENS	TAR-DB2-OPEN-COUNT	Number of SQL open calls
#-ISRT	TAR-DB2-INSERT-COUNT	Number of SQL insert calls
#-DELETE	TAR-DB2-DELETE-COUNT	Number of SQL delete calls
#-UPD	TAR-DB2-UPDATE-COUNT	Number of SQL update calls
#-DDL	TAR-DB2-DDL-COUNT	Number of SQL Data Definition Language calls
#-DYN	TAR-DB2-DYNAMIC-COUNT	Number of SQL dynamic calls
#-CTRL	TAR-DB2-CONTROL-COUNT	Number of SQL control-type calls
#-OTHER	TAR-DB2-OTHER-COUNT	Number of other SQL calls
#-REFS	TAR-NO-REFERENCES	Number of transactions that referenced DB2
$-READ	TAR-DB2-READ-CHG	Charge for number of read-type calls
$-UPDATE	TAR-DB2-UPDATE-CHG	Charge for number of update-type calls
$-DDL	TAR-DB2-DDL-CHG	Charge for number of Data Definition Language calls
$-DYN	TAR-DB2-DYNAMIC-CHG	Charge for number of dynamic SQL calls
$-CTRL	TAR-DB2-CONTROL-CHG	Charge for number of control-type calls
$-OTHER	TAR-DB2-OTHER-CHG	Charge for number of other DB2 calls

Table 13-4 TAR Data Elements for PRSPRINT (Part 6 of 6)

Label	Data Element Name	Definition
Note: The following data elements are in the trailer segment types that may be in the transaction accounting record of a **summarized IRUF**.		
SEGTYPE	TAR-SEGMENT-RES-TYPE	Response segment
R-TRNCAT	TAR-TRANS-CATEGORY	Transaction category
R-RESPTM	TAR-RESPONSE-TIME	Response time work area
R-RESPCNT	TAR-RESPONSE-COUNT	Response count work area
R-RSPTM1	TAR-T (1)	Response threshold 1
R-RSPCT1	TAR-C (1)	Transaction count 1
R-RSPTM2	TAR-T (2)	Response threshold 2
R-RSPCT2	TAR-C (2)	Transaction count 2
R-RSPTM3	TAR-T (3)	Response threshold 3
R-RSPCT3	TAR-C (3)	Transaction count 3
R-RSPTM4	TAR-T (4)	Response threshold 4
R-RSPCT4	TAR-C (4)	Transaction count 4
R-RSPTM5	TAR-T (5)	Response threshold 5
R-RSPCT5	TAR-C (5)	Transaction count 5
R-RSPTM6	TAR-T (6)	Response threshold 6
R-RSPCT6	TAR-C (6)	Transaction count 6
R-RSPTM7	TAR-T (7)	Response threshold 7
R-RSPCT7	TAR-C (7)	Transaction count 7
R-RSPTM8	TAR-T (8)	Response threshold 8
R-RSPCT8	TAR-C (8)	Transaction count 8
R-RSPTM9	TAR-T (9)	Response threshold 9
R-RSPCT9	TAR-C (9)	Transaction count 9
R-RSPT10	TAR-T (10)	Response threshold 10
R-RSPC10	TAR-C (10)	Transaction count 10
R-RSPT11	TAR-T (11)	Response threshold 11
R-RSPC11	TAR-C (11)	Transaction count 11
R-RSPT12	TAR-T (12)	Response threshold 12
R-RSPC12	TAR-C (12)	Transaction count 12

Program Accounting Record (PAR) Data Elements for the PRSPRINT Report

Table 13-5 shows the PAR data elements for the labels in the IRUF report created by the PRSPRINT utility. For more information about the PAR record, see Appendix B, "IRUF Record Layout Descriptions."

Table 13-5 PAR Data Elements for PRSPRINT (Part 1 of 2)

Label	Data Element Name	Definition
SUBTYPE	PAR-RECORD-TYPE	Level of data in this record
VERSION	PAR-RECORD-VERSION	IRUF release identifier
SYSID	PAR-SYSTEM-ID	Originating system
IMS-LEVL	PAR-IMS-LEVEL	IMS release level
PSBNAME	PAR-PSB-NAME	Program name
REGION	PAR-REGION-ID	Processing region name
PSBTYPE	PAR-PSB-TYPE	Program execution type
OS-CODE	PAR-OS-CODE	Operating system code
S-DATE	PAR-PSB-START-DATE	Scheduling start date
S-TIME	PAR-PSB-START-TIME	Scheduling start time
E-DATE	PAR-PSB-END-DATE	Program end date
E-TIME	PAR-PSB-END-TIME	Program end time
ELAPSED	PAR-ELAPSED-TIME	Program elapsed run time
SKED-CPU	PAR-PROG-SCHD-CPU-TIME	Scheduling CPU time
OVHD-CPU	PAR-MSG-REG-OVHD-CPU-TIME	Overhead scheduling CPU time
#-PROGS	PAR-NO-PROGRAMS	Number of programs
S-ABENDS	PAR-NO-SYS-ABENDS	Number of system code abends
U-ABENDS	PAR-NO-USER-ABENDS	Number of user code abends
AB-CODE	PAR-COMPLETION-CODE	Last completion code
#-MGU	PAR-MSG-GET-UNIQUES	Number of message get uniques
#-MGN	PAR-MSG-GET-NEXTS	Number of message get nexts
#-MISRT	PAR-MSG-INSERTS	Number of message inserts
#-MPURGE	PAR-MSG-PURGES	Number of purges
#-MOTHR	PAR-MSG-OTHERS	Number of other message calls
A-CORE	PAR-CORE-ALLOCATED	Allocated core
U-CORE	PAR-CORE-USED	Used core
A-KCOREM	PAR-ALLOCATED-KCORE-MINS	Allocated K-core minutes
PSB-SIZE	PAR-PSB-POOL-SIZE	PSB pool requirement
DMB-SIZE	PAR-DMB-POOL-SIZE	DMB pool requirement
IMS-CPU	PAR-CONT-PROG-CPU-TIME	IMS control region CPU time

Table 13-5 PAR Data Elements for PRSPRINT (Part 2 of 2)

Label	Data Element Name	Definition
U-KCOREM	PAR-COREUSED-KCORE-MINS	Used K-core minutes
IMS-SD	PAR-CP-START-DATE	IMS interval start date
IMS-ST	PAR-CP-START-TIME	IMS interval start time
IMS-ED	PAR-CP-END-DATE	IMS interval end date
IMS-ET	PAR-CP-END-TIME	IMS interval end time
SAP FLG	PAR-SAP-FLAG	SAP exit-invoked indicator (Y or N)
U-AREA1	PAR-USER-AREA	User area (first 10 bytes)
U-AREA2	PAR-USER-AREA	User area (last 10 bytes)
AGN	PAR-AGN	Application group number
DB2ID	PAR-DB2-SUBSYS-ID	ID of DB2 subsystem
NBA	PAR-NBA	Fast Path normal buffer allocation
OBA	PAR-OBA	Fast Path overflow buffer allocation
BUHWM	PAR-BUHWM	Fast Path buffer used high water
DB2TRAN	PAR-PROG-ACCESS-DB2	Number of programs scheduled that made DB2 requests

Terminal (LTERM) Accounting Record (LAR) Data Elements for the PRSPRINT Report

Table 13-6 shows the LAR data elements for the labels in the IRUF report created by the PRSPRINT utility. For more information about the LAR record, see Appendix B, "IRUF Record Layout Descriptions."

Table 13-6 LAR Data Elements for PRSPRINT

Label	Data Element Name	Definition
SUBTYPE	LAR-RECORD-TYPE	Level of data in this record
VERSION	LAR-RECORD-VERSION	IRUF release identifier
SYSID	LAR-SYSTEM-ID	Originating system
IMS-LEVL	LAR-IMS-LEVEL	IMS release level
LTERM	LAR-LOG-TERMINAL-NAME	Logical terminal name
TERMTYPE	LAR-TERM-TYPE	Leased, dial-up
OS-CODE	LAR-OS-CODE	Operating system code
CUSTID	LAR-CUSTOMER-ID	Customer ID
CON-TIME	LAR-CONNECT-TIME	Connect time for this LTERM
USER-ID	LAR-USER-ID	Security system user ID
#-TRANS	LAR-NO-TRANS-SUBMITTED	Number of transactions input
TOT-RESP	LAR-TOTAL-RESPONSE	Total response time
#-SESION	LAR-NO-SESSIONS	Number of connect sessions
ON-DATE	LAR-LOGON-DATE	Logon date
ON-TIME	LAR-LOGON-TIME	Logon time
OFF-DATE	LAR-LOGOFF-DATE	Logoff date
OFF-TIME	LAR-LOGOFF-TIME	Logoff time
#-IN	LAR-MP-CHARS-IN	Number of bytes input
#-OUT	LAR-NO-CHARS-OUT	Number of bytes output
#-OTH-IN	LAR-NO-CHARS-OTHER-IN	Number of bytes input to other PCBs
#-OTH-OUT	LAR-NO-CHARS-OTHER-OUT	Number of bytes output to other PCBs
#-MGU	LAR-MSG-GU-COUNT	Number of message get uniques
#-MGN	LAR-MSG-GN-COUNT	Number of message get nexts
#-MISRT	LAR-MSG-IN-COUNT	Number of inserts
#-MPURGE	LAR-MSG-PU-COUNT	Number of purges
#-MOTH	LAR-MSG-OT-COUNT	Number of other message calls

Appendix A MVIMS Log Record Layouts

The IMS log is the input to the MVIMS IMFLEDIT program, and it contains two MVIMS record types:

- transaction record (X'FA')
- program record (X'F9')

This appendix describes the MVIMS records. Assembler record descriptions for the records created by MVIMS are located in BBSAMP members IMETRN, IMEDBT, and IMEPGM.

Note: The transaction record and program record date fields are year 2000–compliant.

If you are running with IMS 5.1, the X'FA' and X'F9' records show dates in the format 00*yydddc*, where *yy* represents the two-digit year, *ddd* represents the Julian date, and *c* represents a packed decimal sign.

If you are running with IMS 6.1 or later, the X'FA' and X'F9' records show dates in the format *yyyydddc*, where *yyyy* represents the four-digit year, *ddd* represents the Julian date, and *c* represents a packed decimal sign.

Any utility parameters that specify a two-digit year recognize 66 through 99 as the years 1966 through 1999, and 00 through 65 as the years 2000 through 2065.

DBCTL Usage Considerations

For DBCTL CICS threads, the CICS unit-of-work ID (UOWID) replaces the ALTERNATE PCB DESTINATION TRANSACTION field at decimal offset +196 of the MVIMS log transaction record (X'FA').

CICS writes a type 110 record to SMF for every CICS transaction. If a transaction accesses IMS data using the DBCTL attachment, an X'FA' log record is produced for that transaction. During the IMFLEDIT process, an IRUF transaction accounting record (TAR) is created for each X'FA' log record. (TAR record layout descriptions are provided in Appendix B.)

If you need more information about the CICS processing, you can use the CICS UOWID to find the SMF 110 record that corresponds to the transaction. Only the first six bytes of the CICS UOWID can be used for comparison to the SMF 110 record.

Note: If the CICS transaction performs more than one commit or allocates more than one PSB during its execution, more than one IRUF or X'FA' record may correspond to the same SMF 110 record.

Transaction Log Record (X'FA')

This section describes the transaction log record (X'FA'). The transaction log record is written to the MVIMS system log when transaction processing is complete.

Record Format – Fixed Segment

Table A-1 describes the fixed segment of the transaction log record.

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 1 of 10)

Element/Description	Format	Length	Offset DEC/HEX
RECORD IDENTIFICATION Code that identifies this record as a transaction record with a default value of X'FA'.	display	1	004 004
RECORD STATUS Code that identifies the type of transaction as a single record.	display	1	005 005
PST NUMBER PST associated with the region in which the transaction was executed.	binary	2	006 006
OS/390 SYSTEM PRODUCT OS/390 system product version and release number in the format vrr. v represents the version number rr represents the release number	display	3	008 008
SMQ FLAG X'01' shared message queue environment	binary	1	011 00B
IMS LEVEL IMS release number and mod level number.	display	4	012 00C
MVIMS LEVEL MVIMS version and release level.	display	4	016 010
IMS SYSTEM IDENTIFICATION Identification code for the IMS system on which this transaction was processed (IMSID).	display	4	020 014
SMF SYSTEM IDENTIFICATION ID code from SMCA for the computing system on which this program executed.	display	4	024 018
MVIMS SYSTEM IDENTIFICATION Identification code for the IMS system on which this transaction was processed (specified as a parameter to MVIMS – SYSID in IMFECP00).	display	1	028 01C

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 2 of 10)

Element/Description	Format	Length	Offset DEC/HEX
LSO FLAG LSO processing option in effect when this transaction was processed. The values correspond to the IMS LSO parameter values (N, Y, X, and S).	display	1	029 01D
TIMER ERROR FLAG Flag that indicates when an error is detected in the CPU timing services.	binary	1	030 01E
TIMER ERROR DISPATCHER TYPE IMS dispatcher type code when the timing error occurred.	binary	1	031 01F
REGION ID Name assigned to the initiator used to schedule the program into a message region (jobname). For DBCTL threads, this field contains the CICS or ODBA address space jobname.	display	8	032 020
ADDRESS SPACE ID ASID for the region in which this transaction was processed.	binary	2	040 028
PERFORMANCE GROUP OS/390 SRM performance group for the region in which this transaction was processed. If Goal Mode is active, this field contains a hexadecimal value of 0000.	binary	2	042 02A
STORAGE AVAILABLE Number of unused 2K blocks available in the region after initial scheduling. For DBCTL threads, this field contains the amount of storage available in the CICS or ODBA address space at the time the PSB is allocated.	binary	2	044 02C
RESERVED	display	2	046 02E
APPLICATION GROUP NAME The application group name assigned by the security system.	display	8	048 030
PSB NAME Name of the PSB that was scheduled for this transaction.	display	8	056 038
TRANSACTION NAME Transaction code identifying this message. This field is blank for a program that does not process an input transaction (for example, some BMPs). For DBCTL CICS threads, this field contains the four-byte CICS transaction code. For SAP transactions, this field contains the SAP report ID, if available, or the SAP transaction code if the SAP exit is installed.	display	8	064 040
EXTERNAL SUBSYSTEM ID One- to four-character DB2 subsystem ID accessed by this transaction.	display	4	072 048
RESERVED	display	4	076 04C
INPUT CLASS Class used to assign the transaction to a specific message region by the IMS scheduler.	binary	1	080 050

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 3 of 10)

Element/Description	Format	Length	Offset DEC/HEX
PROGRAM TYPE Type of program that processed this transaction: B Batch message processing program (BMP) C Conversational (MPP or JMP) D DBCTL CICS thread E CPI-C driven program (TPI) F Fast Path message-driven program (MDP) M Message processing program (MPP) N Fast Path non-message-driven program (NDP) U Fast Path utility (FP)	display	1	081 051
SPECIAL PROCESSING FLAG Special processing characteristics: J Java messge processing transaction (JMP) K Java batch message processing transaction (JBP) O DBCTL ODBA thread Q PWFI transaction (pseudo wait for input) W WFI transaction (wait for input)	display	1	082 052
FLAG X'80' IMRUTRNx (user exit) was called X'40' SQL call was issued X'20' DL/I database accessed X'10' Fast Path database accessed X'08' APPC/IMS transaction X'04' SAP exit pass 1 X'02' SAP exit pass 2	binary	1	083 053
STORAGE USED Amount, in 2K units, of available message region storage used by this program (link-edit length). For DBCTL threads, this field is zero.	binary	2	084 054
TRANSACTION COMPLETION CODE Transaction completion code in the format xxsssuuu. xx flags sss system abend code uuu user abend code	binary	4	086 056
RESERVED	binary	4	090 05A
NUMBER OF DBD SEGMENTS Number of DBD segments in the variable portion of this record.	binary	2	094 05E
TRANSACTION START DATE Julian date when processing of this transaction was started. Format is packed yyyydddF.	packed	4	096 060
TRANSACTION START TIME Time of day when processing of this transaction was started. The time is expressed in hundredths of seconds (hhmmssth).	packed unsigned	4	100 064

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 4 of 10)

Element/Description	Format	Length	Offset DEC/HEX
TRANSACTION END DATE Julian date when processing of this transaction was completed. Format is packed yyyydddF.	packed	4	104 068
TRANSACTION END TIME Time of day when processing of this transaction was completed. The time is expressed in hundredths of seconds (hhmmssth).	packed unsigned	4	108 06C
TRANSACTION ARRIVAL DATE Julian date when this transaction was submitted for processing. For a non–Fast Path transaction, format is packed yyyydddF. For a Fast Path transaction, format is packed decimal 0099999C. For DBCTL CICS threads, this field contains a zero, because arrival date of the input message is available only to CICS. For DBCTL ODBA threads, there is no input arrival date.	packed	4	112 070
TRANSACTION ARRIVAL TIME Time of day when this transaction was submitted for processing. The time is expressed as packed decimal hhmmsstC. (Arrival time is set by IMS and is the time when a transaction is put in the IMS input message queue or BALG after communications and MFS processing are complete.) • For message switch transactions submitted by a non-message-driven BMP or JBP or an explicit CPI-C driven program, this timestamp is zero. • For message switch transactions for which the original input came from a terminal, this timestamp is the arrival time of the original transaction. • For transactions that originate in another system in an MSC environment, this timestamp is set to the transaction start time, unless the clock of the MSC system is synchronized with the clock of the local IMS. The MSCCLOCK parameter in BBPARM member IMFECP00 is used to specify whether or not MSC systems clocks are synchronized with local time. (For more information, see the MSCCLOCK description on page 7-9.) • For Fast Path transactions, this timestamp is the input queue time in milliseconds. For DBCTL CICS threads, this field contains a zero, because arrival time of the input message is available only to CICS. For DBCTL ODBA threads, there is no input arrival time.	packed	4	116 074
LOGICAL TERMINAL NAME Name of the logical terminal used to submit this transaction. For DBCTL CICS threads, this field contains the four-byte CICS terminal identifier. For APPC/IMS programs, this field contains the partner LU name.	display	8	120 078

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 5 of 10)

Element/Description	Format	Length	Offset DEC/HEX
SECURITY USER ID User ID field, usually from the I/O PCB. The ID is used in connection with signon security. This field contains one of the following values: • user's identification from the source terminal if signon is active. • LTERM name of the source terminal if signon is not active for that terminal. • ID of the user who submitted a BMP or JBP if user ID or LTERM information is not available in the I/O PCB. • blanks if none of the above is true. For DBCTL CICS threads, this field contains the CICS user ID. For APPC/IMS programs, the user ID is passed to IMS by APPC.	display	8	128 080
VTAM NODE NAME Name of the node of the physical terminal from which this transaction was submitted if it is a VTAM terminal. This field is not filled in for Fast Path message-driven transactions. For APPC/IMS transactions, it contains the partner LU name.	display	8	136 088
PHYSICAL LINE NUMBER Line number of the physical terminal from which this transaction was submitted.	binary	4	144 090
PHYSICAL TERMINAL NUMBER Terminal number of the physical terminal from which this transaction was submitted.	binary	4	148 094
PROGRAM NAME For a BMP or JBP, a CPI-C driven program (TPI), or DBCTL thread, this field can be different from the PSB name. For a TPI, the program name is the same as the transaction name.	display	8	152 098
ROUTING CODE Routing code used to schedule this transaction to a BALG (balancing group) for a Fast Path message-driven region.	display	8	160 0A0
FAST PATH FLAG Fast Path characteristics of this transaction if it accessed Fast Path databases: W This transaction waited for the OBA buffer latch.	display	1	168 0A8
OTMA FLAG X'80' OTMA transaction	binary	1	169 0A9
FAST PATH BUFFERS USED If this transaction accessed Fast Path databases, this field shows how many Fast Path buffers were actually used.	binary	2	170 0AA
NORMAL BUFFER ALLOCATION If this transaction ran in a region that supported Fast Path database access, this value is the NBA value from the JCL.	binary	2	172 0AC

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 6 of 10)

Element/Description	Format	Length	Offset DEC/HEX
OVERFLOW BUFFER ALLOCATION If this transaction ran in a region that supported Fast Path database access, this value is the OBA value from the JCL.	binary	2	174 0AE
FAST PATH CI CONTENTIONS Number of Fast Path CI contentions that occurred in the transaction sync interval.	binary	2	176 0B0
FAST PATH WAIT FOR BUFFERS Number of times this transaction had to wait for a Fast Path buffer to become available.	binary	2	178 0B2
SYNC POINT RETURN CODE Result of sync point processing for a transaction that accessed Fast Path databases.	binary	2	180 0B4
BALANCING GROUP Q COUNT For a Fast Path message-driven transaction, this value is the number of transactions queued on the same balancing group when this transaction went through sync point processing.	binary	2	182 0B6
MESSAGE QUEUE RELATIVE RECORD NUMBER Input message DRRN.	display	4	184 0B8
MSC TRACE For an MSC transaction, this value is the last two bytes of PSTMST, tracing the system IDs.	binary	2	188 0BC
MSC INPUT SOURCE Code that identifies MSC origin. X'08' Response message was inserted. X'20' Terminal input. X'40' Transaction arrival date/time are set to arrival date/time from originating MSC system due to clock synchronization. (For more information, see the MSCCLOCK description on page 7-9.) X'80' Message switch.	display	2	190 0BE
MSC INPUT SYSID In an MSC environment, this value is the system ID of the system where this transaction was entered.	binary	1	192 0C0
MSC DESTINATION SYSID – I/O PCB In an MSC environment, this value is the system ID of the system to which output is sent through the I/O PCB.	binary	1	193 0C1
MSC DESTINATION SYSID – MESSAGE SWITCH In an MSC environment, this value is the system ID of the system to which a message switch transaction was inserted (the last one if more than one).	binary	1	194 0C2
MSC DESTINATION SYSID – ALTERNATE PCB In an MSC environment, this value is the system ID of the system to which output to an LTERM was inserted through an alternate PCB (the last one if more than one).	binary	1	195 0C3

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 7 of 10)

Element/Description	Format	Length	Offset DEC/HEX
This field has two possible uses: ALTERNATE PCB DESTINATION TRANSACTION Transaction code of the last message switch transaction inserted by this transaction, if any. CICS UNIT OF WORK ID This element is for DBCTL CICS threads only. The first six bytes of the CICS UOW (unit of work) ID can be used to match the first six bytes of the UOWID in the CICS SMF 110 record.	display	8	196 0C4
ALTERNATE PCB DESTINATION LTERM LTERM name of the destination of an alternate PCB message inserted by this transaction, if any.	display	8	204 0CC
MSC ARRIVAL DATE Julian date when this MSC transaction was submitted for processing in the originating system. Format is packed decimal yyyydddF. **Warning:** Clocks cannot be synchronized with the processing system. Note that MSC arrival date is not affected by the MSCCLOCK parameter in BBPARM member IMFECP00.	packed	4	212 0D4
MSC ARRIVAL TIME Time of day when this MSC transaction was submitted for processing in the originating system. Format is packed decimal hhmmsstC. **Warning:** Clocks cannot be synchronized with the processing system. Note that MSC arrival time is not affected by the MSCCLOCK parameter in BBPARM member IMFECP00.	packed	4	216 0D8
TRANSACTION LOGGING IDENTIFIER Identifier of the MVIMS module that logged this transaction record.	display	4	220 0DC
CONTROL REGION DLI TIME Amount of control region CPU time, expressed in microseconds, used by the DL/I Analyzer to process the transaction. Does not include database open/close time. Bit 51 is equal to one microsecond. If the buffer handler option is turned off, this time also includes buffer handler CPU time. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	224 0E0
CONTROL REGION BUFFER HANDLER TIME Amount of control region CPU time, expressed in microseconds, used by the buffer handler module during the processing of the transaction. Bit 51 is equal to one microsecond. The amount is zero if the buffer handler option (BHTO) is turned off. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	232 0E8

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 8 of 10)

Element/Description	Format	Length	Offset DEC/HEX
CONTROL REGION OPEN CLOSE CPU TIME Amount of control region CPU time, expressed in microseconds, used by the DL/I Analyzer to open and close required databases while processing this transaction. Bit 51 is equal to one microsecond. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	240 0F0
DEPENDENT REGION DLI TIME Amount of dependent region CPU time, expressed in microseconds, used by the DL/I Analyzer to process the transaction. Bit 51 is equal to one microsecond. If the buffer handler option is turned off, this time also includes buffer handler CPU time.	binary	8	248 0F8
DEPENDENT REGION BUFFER HANDLER TIME Amount of dependent region CPU time, expressed in microseconds, used by the buffer handler module during the processing of this transaction. The amount is zero if the buffer handler option (BHTO) is turned off.	binary	8	256 100
DEPENDENT REGION OPEN CLOSE CPU TIME Amount of dependent region CPU time, expressed in microseconds, used by the DL/I Analyzer to open and close databases for this transaction. Only Fast Path DEDBs are opened from the dependent region.	binary	8	264 108
DEPENDENT REGION CPU TIME Amount of CPU time, expressed in microseconds, used by the application program to process this transaction. Bit 51 is equal to one microsecond. The amount includes DB2 CPU time if the parameter FEATURE=NODB2 is specified in PARMLIB member IMFSYS00 (see PARMLIB member IMFSYSBB for more information).	binary	8	272 110
DLISAS REGION DLI TIME If LSO=S, the amount of DLISAS region CPU time, expressed in microseconds, used by the DL/I Analyzer to process the transaction. Does not include database open/close time. Bit 51 is equal to one microsecond. If the buffer handler option (BHTO) is turned off, this time also includes buffer handler CPU time. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	280 118
DLISAS BUFFER HANDLER CPU TIME If LSO=S, the amount of DLISAS region CPU time, expressed in microseconds, used by the Buffer Handler module during the processing of the transaction. Bit 51 is equal to one microsecond. The amount is zero if the buffer handler option (BHTO) is turned off. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	288 120

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 9 of 10)

Element/Description	Format	Length	Offset DEC/HEX
DLISAS OPEN CLOSE CPU TIME If LSO=S, the amount of DLISAS region CPU time, expressed in microseconds, used by the DL/I Analyzer to open and close required databases while processing this transaction. Bit 51 is equal to one microsecond. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	296 128
DEPENDENT REGION DB2 CPU TIME Amount of CPU time, expressed in microseconds, used in the dependent region to process normal and service calls to DB2.	binary	8	304 130
MESSAGE GET UNIQUE COUNT Number of message get unique calls issued by the transaction (does not include an MGU with a QC status code).	binary	4	312 138
MESSAGE GET NEXT COUNT Number of message get next calls issued by the transaction.	binary	4	316 13C
MESSAGE INSERT COUNT Number of message insert calls issued by the transaction.	binary	4	320 140
MESSAGE PURGE COUNT Number of message purge calls issued by the transaction.	binary	4	324 144
MESSAGE OTHER COUNT Number of other-type calls (checkpoint, statistics, System Service, and so on) issued by the transaction.	binary	4	328 148
INPUT CHARACTER COUNT Number of characters in this input transaction (after MFS formatting).	binary	4	332 14C
SPA INPUT CHARACTER COUNT Number of input characters from a scratch pad area during transaction processing.	binary	4	336 150
OUTPUT CHARACTER COUNT Number of output characters to the originating logical terminal during transaction processing (before MFS formatting).	binary	4	340 154
SPA OUTPUT CHARACTER COUNT Number of output characters to a scratch pad area during transaction processing.	binary	4	344 158
MESSAGE SWITCH OUTPUT CHAR COUNT Number of output characters to the message input queue as a result of message switches by this transaction.	binary	4	348 15C
ALTERNATE TERM OUTPUT CHAR COUNT Number of output characters for this transaction to an LTERM other than the originating logical terminal (before MFS formatting).	binary	4	352 160
SYNC BUFFER FLUSH KEY WRITE	binary	2	356 164
SYNC BUFFER FLUSH NONKEY WRITE	binary	2	358 166

Table A-1 Transaction Log Record Layout – Fixed Segment (Part 10 of 10)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED USER Area reserved for user functions or additions.	display	16	360 168
UTC ADJUSTMENT VALUE IMS Universal Time Clock (UTC) adjustment value. This element is a UTC-to-local time adjustment value. It applies only with IMS 6.1 or later.	packed	2	376 178
IMS TRANSACTION CODE IMS transaction code for SAP applications if the SAP exit pass 1 (TRNSAP1) flag is on (at displacement 083 053).	display	8	378 17A
RESERVED	display	208	386 182
WORKLOAD MANAGER SERVICE CLASS Workload manager service class of the transaction.	display	8	594 252
SMQ GROUP NAME Shared message queues group name.	display	8	602 25A
UOW FOR TRANSACTION	display	34	610 262
RESERVED	display	120	644 284
COUNT OF CHECKPOINT/SYNCPOINTS Number of checkpoints/syncpoints taken during the execution of this transaction.	binary	4	764 2FC
MAXIMUM LOCKS HELD Maximum number of locks held between any two checkpoints/syncpoints.	binary	4	768 300
TOTAL LOCKS Total number of locks held for all checkpoints. (Divide this value by the number of checkpoints/syncpoints to obtain the average number of locks per checkpoint.)	binary	4	772 304
FIXED RECORD LENGTH End of the fixed portion of the record, which can be followed by 0 to 500 database segments.			776 308

Record Format – DBD Variable Segment

This section describes the DBD variable segments of the transaction log record. The DBD variable segments contain measurement data on database activity. There can be multiple occurrences of this segment, up to a maximum of 500.

Note: When the counter value is too large for its field in the database trailer segment, an overflow occurs. For any overflow, the overflow counter index indicates the offset of the element that overflowed into the overflow fullword counter. If more than one element overflowed or if the first element overflowed into the overflow fullword counter, an additional database trailer segment is created to contain the overflow. This additional segment is pointed to by the overflow pointer at decimal offset 015.

The additional database trailer segment contains the final tally of all the elements that overflowed unless overflow occurs again. If overflow occurs again, the overflow fullword counter in the additional segment indicates the first element within the segment that overflowed. Any additional overflows are controlled similarly.

Table A-2 provides detailed information about the DBD variable segment of the transaction log record.

Table A-2 Transaction Log Record Layout – DBD Variable Segment (Part 1 of 4)

Element/Description	Format	Length	Offset DEC/HEX
DBD SEGMENT Database segments are maintained for each DBD referenced in processing this transaction. The maximum number of segments is defined by the DBT parameter in BBPARM (see Appendix C, "How Product Libraries Should Be Used").	group	48	000 000
DBD NAME Name of the database. If the maximum number of segments is exceeded, an overflow segment is created with OTHERS in this field. If the Event Collector DBIO data collection parameter is IOWAITS, a special segment, ALLDBS, can be created to account for sync point write activity.	display	8	000 000

Table A-2 Transaction Log Record Layout – DBD Variable Segment (Part 2 of 4)

Element/Description	Format	Length	Offset DEC/HEX
ORGANIZATION TYPE Code that identifies the type of organization used for this database. X'00' no organization, used for ALLDBS segment X'44' root index X'20' HDAM X'64' HDAM plus root index X'10' HIDAM X'54' HIDAM plus root index X'74' HIDAM Index X'08' HISAM-CASE-2 X'04' HISAM X'02' HSAM X'01' SSAM X'05' SHISAM X'C4 (D) DEDB X'D1' (J) MSDB, nonrelated, key in segment X'C1' (A) MSDB, nonrelated, LTERM key X'C3' (C) MSDB, related, fixed X'C7' (G) MSDB, related, dynamic X'80' DB2 trailer format	binary	1	008 008
DBTFLAG1 Special database flag. X'01' secondary DBT X'02' catch-all DBT X'08' ddname DBT X'40' Some DBT CNTRS have overflowed to OTHERS. For OTHERS segments, some count has been lost X'80' DBT overflow has occurred. Another DBT may exist.	binary	1	009 009
DMB POOL SIZE REQUIREMENT Amount of DMB buffer pool space required by this database, expressed in bytes.	binary	2	010 00A
DBTFLAG2 Special database flag. X'04' HALDB (partitioned database) X'08' OSAM sequential buffering in effect X'80' VSAM	binary	1	012 00C
DBTFLAG3 Special database flag. X'02' DBTSECNO wrapped X'04' DBTDDNNO wrapped	binary	1	013 00D
DD POINTER Chain of pointers for the DD segments for this database. If the X'04' bit in DBTFLAG3 is set, X'100' is added to this value.	binary	1	014 00E

Table A-2 Transaction Log Record Layout – DBD Variable Segment (Part 3 of 4)

Element/Description	Format	Length	Offset DEC/HEX
OVERFLOW POINTER Overflow pointer to next overflow segment. If the X'02' bit in DBTFLAG3 is set, X'100'is added to this value.	binary	1	015 00F
GET UNIQUE COUNT Number of times a get unique was requested in processing this transaction.	binary	2	016 010
GET NEXT COUNT Number of times a get next was requested in processing this transaction.	binary	2	018 012
INSERT COUNT Number of times an insert was requested in processing this transaction.	binary	2	020 014
DELETE COUNT Number of times a delete was requested in processing this transaction.	binary	2	022 016
REPLACE COUNT Number of times a replace was requested in processing this transaction.	binary	2	024 018
OTHER COUNT Number of system service calls issued against this database by the transaction.	binary	2	026 01A
KEY READS Number of reads issued to the key sequence data area in satisfying the DL/I requests.	binary	2	028 01C
KEY WRITES Number of writes issued to the key sequence data area in satisfying the DL/I requests.	binary	2	030 01E
NONKEY READS Number of reads issued to the OSAM or entry sequence data area in satisfying the DL/I requests.	binary	2	032 020
NONKEY WRITES Number of writes issued to the OSAM or entry sequence data area in satisfying the DL/I requests.	binary	2	034 022
NONKEY WRITES Number of writes issued to the OSAM or entry sequence data area in satisfying the DL/I requests.	binary	2	034 022
NO IO Number of times the requested data was found in the I/O buffer.	binary	2	036 024
NO IO ALTERS Number of alters to an already altered buffer.	binary	2	038 026
SERIALIZED IO Number of I/Os that occurred under a serialized IMS task (in the control region or DLISAS region) instead of under the dependent region task.	binary	2	040 028

Table A-2 Transaction Log Record Layout – DBD Variable Segment (Part 4 of 4)

Element/Description	Format	Length	Offset DEC/HEX
OVERFLOW COUNTER INDEX If a counter overflow occurred, this element is the index to the counter affected (hexadecimal offset of the elements in the DBD variable segment that overflowed into the overflow fullword counter at decimal offset 044).	binary	2	042 02A
OVERFLOW FULLWORD COUNTER If a counter overflow occurred, this fullword is used to hold this larger value.	binary	4	044 02C

Record Format – DB2 Variable Segment (Optional)

This section describes the optional DB2 variable segment of the transaction log record. There is, at most, one occurrence of this segment. The segment will not be produced if the parameter FEATURE=NODB2 is specified in PARMLIB member IMFSYS00 (see PARMLIB member IMFSYSBB for more information).

Table A-3 provides detailed information about the DB2 variable segment of the MVIMS transaction log record.

Table A-3 Transaction Log Record Layout – DB2 Variable Segment (Part 1 of 2)

Element/Description	Format	Length	Offset DEC/HEX
DB2 SEGMENT This element redefines the **DBD SEGMENT**. If ORGANIZATION TYPE for the DBD is DB2 (X'80'), there is only one record per transaction.	group	48	000 000
PLAN NAME DB2 plan name for this transaction.	group	8	000 000
ORGANIZATION TYPE Following type is added: X'80' Special segment for DB2 calls	display	1	008 008
RESERVED AREA	binary	3	009 009
SELECT/FETCH CALLS Number of SQL SELECT and FETCH calls made to DB2.	display	4	012 00C
OPEN CALLS Number of SQL OPEN calls made to DB2.	binary	4	016 010
INSERT CALLS Number of SQL INSERT calls made to DB2.	binary	4	020 014
DELETE CALLS Number of SQL DELETE calls made to DB2.	binary	4	024 018

Table A-3 Transaction Log Record Layout – DB2 Variable Segment (Part 2 of 2)

Element/Description	Format	Length	Offset DEC/HEX
UPDATE CALLS Number of SQL UPDATE calls made to DB2.	binary	4	028 01C
DDL CALLS Number of SQL Data Definition Language calls made to DB2 (CREATE, DROP, ALTER, COMMENT and LABEL).	binary	2	032 020
DYNAMIC CALLS Number of SQL dynamic calls made to DB2 (PREPARE, DESCRIBE and EXECUTE).	binary	2	034 022
CONTROL CALLS Number of SQL control type calls made to DB2 (GRANT, REVOKE).	binary	2	036 024
OTHER CALLS Number of other SQL control type calls made to DB2 (EXPLAIN, LOCK, LABEL, CLOSE).	binary	2	038 026
RESERVED AREA	binary	8	040 028

Program Log Record (X'F9')

Table A-4 describes the program log record (X'F9'). This record is written to the IMS log at the time the program is terminated. The table provides detailed information about the program log record layout.

Table A-4 Program Log Record Layout (Part 1 of 6)

Element/Description	Format	Length	Offset DEC/HEX
RECORD IDENTIFICATION Code that identifies this record as a program accounting record with a default value of X'F9'.	display	1	004 004
RESERVED	display	1	005 005
PST NUMBER PST associated with the region in which the program was scheduled.	binary	2	006 006
OS/390 SYSTEM PRODUCT OS/390 system product version and release number in the format vrr. v represents the version number rr represents the release number	display	3	008 008
RESERVED	display	1	011 00B
IMS LEVEL IMS release number and mod level number.	display	4	012 00C

Table A-4 Program Log Record Layout (Part 2 of 6)

Element/Description	Format	Length	Offset DEC/HEX
MVIMS LEVEL MVIMS version and release level.	display	4	016 010
IMS SYSTEM IDENTIFICATION IMS system on which this program was executed (IMSID).	display	4	020 014
SMF SYSTEM IDENTIFICATION ID code from SMCA for the computing system on which this program was executed.	display	4	024 018
MVIMS SYSTEM IDENTIFICATION IMS system on which this program executed (specified as a parameter to MVIMS – SYSID in IMFECP00).	display	1	028 01C
LSO FLAG LSO processing option in effect when this program was executed. The values correspond to the IMS LSO parameter values: N, Y, X, or S.	display	1	029 01D
PROCESSING OPTION FLAGS 1 and 2 Event Collector data collection parameter options set in IMFECP00 that are in effect during processing. X'0200' RGNIOPT=CONTINUE X'0010' BHTO=ON X'0004' DEPREC=NO	binary	2	030 01E
REGION ID Name assigned to the initiator used to schedule the program into a message region (jobname). For DBCTL threads, this field contains the CICS or ODBA address space jobname.	display	8	032 020
ADDRESS SPACE ID ASID for the region in which this program was executed.	binary	2	040 028
PERFORMANCE GROUP OS/390 SRM performance group for the region in which this program was executed. If Goal Mode is active, this field contains a hexadecimal value of 0000.	binary	2	042 02A
STORAGE AVAILABLE Size of the message region, expressed in 2K units, available after initial scheduling. For DBCTL threads, this field contains the amount of storage available in the CICS or ODBA address space at the time the PSB is allocated.	binary	2	044 02C

Table A-4 Program Log Record Layout (Part 3 of 6)

Element/Description	Format	Length	Offset DEC/HEX
PROCESSING OPTION FLAGS 1 and 2 Event Collector data collection parameter options set in IMFECP00 that are in effect during processing. X'2000' CPU=NONE X'0200' CPU=DEPPGM | DEPDB2 | DEP X'0040' BILLOVHD=YES X'0020' RNSYNC=YES X'0010' BMP=NO X'0008' BIO=IOWAITS X'0004' DBIO=NONE	binary	2	046 02E
APPLICATION GROUP NAME Application group name assigned by the security system.	display	8	048 030
PROGRAM NAME Name of the program (PSB) that processed this transaction. If FINALPGM, this value is a special record produced at the end of an IMS session. It contains the remaining CPU overhead values.	display	8	056 038
TRANSACTION NAME Transaction code of the last transaction processed by this program.	display	8	064 040
EXTERNAL SUBSYSTEM ID One- to four-character DB2 subsystem ID accessed by this program.	display	4	072 048
RESERVED	display	4	076 04C
INPUT CLASS Input class of the last transaction processed by this program.	binary	1	080 050
PROGRAM TYPE B Batch message processing program (BMP) C Conversational (MPP and JMP) D DBCTL CICS thread E CPI-C driven program (TPI) F Fast Path message-driven program (MDP) M Message processing program (MPP) N Fast Path non-message-driven program (NDP) S Special (MVIMS FINALPGM overhead record) U Fast Path utility (FPU)	display	1	081 051
ADDITIONAL PROGRAM TYPES J Java message processing (JMP) K Java batch message processing (JBP) O DBCTL ODBA thread	display	1	082 052
FLAG X'80' IMEUPGMx (user exit) called X'40' SQL call was issued X'08' APPC/IMS transaction X'04' SAP exit called	binary	1	083 053

Table A-4 Program Log Record Layout (Part 4 of 6)

Element/Description	Format	Length	Offset DEC/HEX
STORAGE USED Amount, in 2K units, of available message region storage used by this program (link-edit length). For DBCTL threads, this field is zero.	binary	2	084 054
PROCESSING OPTION FLAGS 5 and 6	binary	2	086 056
WORKLOAD MANAGER SERVICE CLASS Workload manager service class of the program.	display	8	088 058
PROGRAM START DATE Julian date when this program was initiated. Format is packed yyyydddF.	packed	4	096 060
PROGRAM START TIME Time of day when this program was initiated. The time is expressed in hundredths of seconds (hhmmssth).	packed unsigned	4	100 064
PROGRAM END DATE The Julian date when processing of this program was terminated. Format is packed yyyydddF.	packed	4	104 068
PROGRAM END TIME The time of day when processing of this program was terminated. The time is expressed in hundredths of seconds (hhmmssth).	packed unsigned	4	108 06C
NORMAL BUFFER ALLOCATION If this program was scheduled in a region that supported Fast Path database access, this value is the NBA value from the JCL.	binary	2	112 070
OVERFLOW BUFFER ALLOCATION If this program was scheduled in a region that supported Fast Path database access, this value is the OBA value from the JCL.	binary	2	114 072
FAST PATH BUFFER HIGH WATER MARK Maximum number of Fast Path buffers used for all transactions processed during this scheduling.	binary	2	116 074
PROGRAM COMPLETION CODE Program completion code in format xxsssuuu. xx flags sss system abend code uuu user abend code	binary	4	120 078
PSB POOL REQUIREMENT Amount of PSB buffer pool space required by this program, expressed in units of 8 bytes.	binary	2	124 07C
DMB POOL REQUIREMENT Amount of DMB buffer pool space required by this program, expressed in bytes.	binary	2	126 07E
PROGRAM LOGGING IDENTIFIER MVIMS module that logged this program record.	binary	4	128 080

Table A-4 Program Log Record Layout (Part 5 of 6)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED	display	4	132 084
CONTROL REGION SCHEDULING CPU TIME Amount of CPU time used in the control region in scheduling and terminating processing for this program. This time is expressed in microseconds, with bit 51 equal to one microsecond. This time can include DB close activity. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	136 088
CONTROL REGION OVERHEAD CPU TIME CPU overhead time in the control region since the last program record (from any region). This time is expressed in microseconds, with bit 51 equal to one microsecond. **Note:** This time is *not* attributable only to this program; it includes the time for such functions as MFS and queueing.	binary	8	144 090
DEPENDENT REGION OVERHEAD CPU TIME CPU overhead time since the last program record in this region. This time is expressed in microseconds, with bit 51 equal to one microsecond, and it includes scheduling time. For DBCTL threads, this field contains a zero.	binary	8	152 098
DEPENDENT REGION PROGRAM LOAD TIME	binary	8	160 0A0
DLISAS SCHEDULING CPU TIME Amount of CPU time used in the DLISAS region in scheduling and terminating processing for this program. This time is expressed in microseconds, with bit 51 equal to one microsecond. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	168 0A8
DLISAS REGION OVERHEAD CPU TIME CPU overhead time in the DLISAS region since the last program record (from any region). This time is expressed in microseconds, with bit 51 equal to one microsecond. **Note:** This time is *not* attributable only to this program; it includes the time for such functions as MFS and queueing.	binary	8	176 0B0
CPU TIMER WORKING STORAGE	binary	8	184 0B8
MESSAGE GET UNIQUES Number of message get uniques issued by this program in processing the transactions. This count includes messages with a QC status. (Therefore, the count is usually one more than the number of transactions processed.)	binary	4	192 0C0
MESSAGE GET NEXTS Number of message get nexts issued by the program in processing the transactions.	binary	4	196 0C4
MESSAGE INSERTS Number of message inserts issued by the program in processing the transactions.	binary	4	200 0C8

Table A-4 Program Log Record Layout (Part 6 of 6)

Element/Description	Format	Length	Offset DEC/HEX
MESSAGE PURGES Number of message purges issued by the program in processing the transactions.	binary	4	204 0CC
MESSAGE OTHERS Number of other message calls including system service calls.	binary	4	208 0D0
RESERVED	display	8	212 0D4
RESERVED USER Reserved for user functions.	display	16	220 0DC
RESERVED	display	4	236 0EC
END OF RECORD Program record is a fixed length record.			240 0F0

Appendix B IRUF Record Layout Descriptions

This appendix contains the IRUF record layout descriptions for the transaction accounting record (TAR), the program accounting record (PAR), and the terminal (LTERM) accounting record (LAR).

The formats for IRUF transaction, program, and terminal records are distributed in BBSAMP as follows:

Member Name	Format
CIRUFR02	Assembler (all record types)
CIMTAR02	COBOL transaction accounting records
CIMPAR01	COBOL program accounting records
CIMLAR01	COBOL terminal accounting records
SASIRUF	SAS (all record types)

DBCTL Usage Considerations

For DBCTL CICS threads, the CICS unit-of-work ID (UOWID) replaces the ALTERNATE PCB DESTINATION TRANSACTION field at decimal offset +144 of the IRUF transaction accounting record (TAR).

CICS writes a type 110 record to SMF for every CICS transaction. If a transaction accesses IMS data using the DBCTL attachment, a transaction log record (X'FA') is produced for that transaction. During the IMFLEDIT process, an IRUF TAR record is created for each X'FA' log record.

If you need more information about the CICS processing, you can use the CICS UOWID to find the SMF 110 record that corresponds to the transaction. Only the first six bytes of the CICS UOWID can be used for comparison to the SMF 110 record.

Note: If the CICS transaction performs more than one commit or allocates more than one PSB during its execution, more than one IRUF or X'FA' record may correspond to the same SMF 110 record.

Transaction Accounting Record Layout

This section describes the layout of the transaction accounting record (TAR). This record contains a fixed segment, a DBD variable segment, an optional DB2 variable segment, and a response variable segment. A terminal segment is included in the fixed segment portion.

Note: If you are using the MVIMS version 3.2 IRUF format rather than the version 3.3 format (see page 10-8), the terminal segment is in the variable portion of the TAR, rather than the fixed portion. If you want to see the TAR layout for the version 3.2 IRUF format, refer to the version 3.2 *MainView for IMS Offline Products Reference Manual* or mapping DSECTS CIRUFR01 or CIMTAR01.

Record Format – Fixed Segment

Table B-1 shows the record format for the fixed segment of the transaction accounting record (TAR).

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 1 of 13)

Element/Description	Format	Length	Offset DEC/HEX
RECORD IDENTIFICATION Code that identifies this record as a transaction accounting record (value T).	display	1	000 000
RESERVED AREA	display	1	001 001
RECORD VERSION MVIMS IRUF record version code: X'01' MVIMS version 3.2.00 or earlier X'02' MVIMS version 3.3.00 or later	display	2	002 002
TRANSACTION IDENTIFICATION	**group**	**32**	**004 004**
CUSTOMER ID User-specified identification that relates this transaction to a specific user. This field is built by the Log Edit user exit routine. The default value is LTERM.	display	18	004 004
TRANSACTION CODE Transaction code that identifies this message. If *LINECHG, this value is a dummy transaction record created during IRUF summarization per MVIMS to contain terminal connect time charges. For SAP transactions (SAP EXIT USED is Y), this field contains the SAP report ID, if available, or the SAP transaction code. For DBCTL CICS threads, this field contains the four-byte transaction code.	display	8	022 016

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 2 of 13)

Element/Description	Format	Length	Offset DEC/HEX
APPC/IMS FLAG N non-APPC/IMS transaction Y APPC/IMS transaction O OTMA transaction	display	1	030 01E
SYSTEM IDENTIFICATION Identification code for the IMS system on which this transaction was processed (specified as a parameter to BBPARM member IMFECP00).	display	1	031 01F
IMS LEVEL IMS release level under which this transaction was processed.	display	4	032 020
TRANSACTION CHARACTERISTICS	**group**	**24**	**036 024**
PROGRAM NAME Name of the program (PSB) that processed this transaction.	display	8	036 024
REGION ID Name assigned to the initiator used to schedule the program into a message region. For DBCTL threads, this field contains the CICS or ODBA address space jobname.	display	8	044 02C
INPUT CLASS Class used to assign the program to a specific message region by an initiator.	display	3	052 034
RECORD TYPE Type of information stored in this record: D Detail level data S Summary level data	display	1	055 037
TRANSACTION TYPE Transaction processing mode: blank Standard – message processing transaction (MPP) B Batch message processing transaction (BMP) C Conversational (MPP or JMP) D DBCTL CICS thread E CPI-C driven program transaction (TPI) F Fast Path message-driven transaction (MDP) N Fast Path non-message-driven transaction (NDP) U Fast Path utility transaction (FPU)	binary	1	056 038
R OPTION Response option flag. If equal to R, the record was created with the response option in effect and the other R fields are used.	display	1	057 039
DET TRANS CATEGORY (detail) Category code assigned to transaction through user exit routine IMFCEXIT in Log Edit. This code is used to determine which set of response time thresholds is to be used in building the summary response segment in this record.	binary	2	058 03A

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 3 of 13)

Element/Description	Format	Length	Offset DEC/HEX
TIMESTAMPS (detail) Timestamps are either detail or summarized as specified in the RECORD TYPE field. For a detail record, the timestamps contain the dates and times for the time of arrival, start, and stop of a single transaction.	n/a	n/a	n/a
TIMESTAMPS (summarized) For a summarized record, the timestamps identify a time range. Data is collected for the number of transactions processed (NO TRANSACTIONS) and summarized for the time range. The arrival timestamp contains the earliest date and time encountered. The stop timestamp contains the latest date and time encountered. The termination timestamp, reflecting transaction purge time, is not a currently supported data element.	n/a	n/a	n/a
TRANSACTION ARRIVAL DATE Julian date (yyyyddd) when this transaction was submitted. For DBCTL CICS threads, this field is zero, because arrival date of the input message is available only to CICS. For DBCTL ODBA threads, there is no input arrival date.	binary	4	060 03C
TRANSACTION ARRIVAL TIME Time of day, in hundredths of seconds, when a transaction completes communication and message format processing and is placed in the input message queue or BALG. This timestamp is the arrival time of the original transaction for message switch transactions for which the original input came from a terminal. This timestamp is set to the transaction start time for message switch transactions submitted by a non-message-driven BMP or JBP and for the pseudo transaction created for such a BMP, JBP, NDP, or FPU. For transactions submitted from another system in an MSC environment, this timestamp is the arrival time in the processing system with the response option in effect; otherwise, it is the arrival time in the originating system. For DBCTL CICS threads, this field is zero, because arrival time of the input message is available only to CICS. For DBCTL ODBA threads, there is no input arrival time.	binary	4	064 040
TRANSACTION START DATE Julian date (yyyyddd) when processing for this transaction started.	binary	4	068 044
TRANSACTION START TIME Time of day, in hundredths of seconds, when processing for this transaction started.	binary	4	072 048
TRANSACTION STOP DATE Julian date (yyyyddd) when processing of this transaction ended.	binary	4	076 04C
TRANSACTION STOP TIME Time of day, in hundredths of seconds, when processing for this transaction ended.	binary	4	080 050
TRANSACTION TERM DATE Julian date (yyyyddd) when this transaction was purged. This field is currently not supported and is set to the TRANSACTION STOP date.	binary	4	084 054

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 4 of 13)

Element/Description	Format	Length	Offset DEC/HEX
TRANSACTION TERM TIME Time of day, in hundredths of seconds, when processing on this transaction was purged. This field is currently not supported and is set to the TRANSACTION STOP time.	binary	4	088 058
SECURITY IDS	**group**	**24**	**092 05C**
LOGICAL TERMINAL NAME Name of the LTERM that submitted the transaction. A special LTERM name of ??-BLANK is supplied when no terminal input occurs. This value is in the pseudo transaction record that MVIMS creates for • a non-message-driven BMP or JBP (NDP – non-message-driven program) • FPU (Fast Path utility) • any message switch transactions originating from a non-message-driven BMP or JBP For DBCTL CICS threads, this field contains the four-byte CICS terminal identifier. For APPC/IMS programs, this field contains the partner LU name.	display	8	092 05C
SECURITY USER ID User ID field, usually from the I/O PCB. The ID is used in connection with signon security. The field contains one of the following values: • user's identification from the source terminal if signon is active • LTERM name of the source terminal if signon is not active for that terminal • ID of the user who submitted a BMP or JBP if user ID or LTERM information is not available in the I/O PCB • blanks if none of the above are true For APPC/IMS transactions, the user ID is passed to IMS from APPC. This field is equivalent to DB2 AUTHID. For DBCTL CICS threads, this field contains the CICS user ID.	display	8	100 064
AGN Application group name assigned by the security system.	display	8	108 06C
ADDRESS SPACE ID ASID for the region in which this transaction was processed.	binary	2	116 074
PERFORMANCE GROUP OS/390 SRM performance group for the region in which this transaction was processed. If Goal Mode is active, this field contains a hexadecimal value of 0000.	display	2	118 076
IMS SYSTEM IDENTIFICATION Identification code for the IMS system on which this transaction was processed (IMSID).	display	4	120 078

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 5 of 13)

Element/Description	Format	Length	Offset DEC/HEX
SMF SYSTEM IDENTIFICATION ID code from SMCA for the computing system on which this program executed.	display	4	124 07C
VTAM NODE NAME Name of the node of the physical terminal from which this transaction was submitted if it is a VTAM terminal. This field is not filled in for Fast Path message-driven transactions. For APPC/IMS transactions, it contains the partner LU name.	display	8	128 080
SPECIAL PROCESSING FLAG Special LSO processing characteristics in effect for this transaction. This flag is one of the four IMS LSO options (N, Y, X, or S).	display	1	136 088
OS CODE OS/390 operating system under which this transaction was processed.	display	3	137 089
USER AREA MVIMS uses the first 16 bytes of the user area to store the extra data items listed below. This data is not used in any Performance Reporter or Transaction Accountant offline processing, so it can be modified by the user without consequence. If a user exit places user data in the transaction log record, the data will be transferred to this user area.	**group**	**20**	**140 08C**
MSC INPUT SYSID In an MSC environment, this value is the system ID of the system where this transaction was entered.	binary	1	140 08C
MSC DESTINATION SYSID – I/O PCB In an MSC environment, this value is the system ID of the system to which output is sent through the I/O PCB.	binary	1	141 08D
MSC DESTINATION SYSID – MESSAGE SWITCH In an MSC environment, this value is the system ID of the system to which a message switch transaction was inserted (the last one if more than one).	binary	1	142 08E
MSC DESTINATION SYSID – ALTERNATE PCB In an MSC environment, this value is the system ID of the system to which output to an LTERM was inserted through an alternate PCB (the last one if more than one).	binary	1	143 08F
This field has two possible uses: ALTERNATE PCB DESTINATION TRANSACTION Transaction code of the last message switch transaction inserted by this transaction, if any. CICS UNIT OF WORK ID For DBCTL CICS threads only. The first 6 bytes of the CICS UOW (unit of work) ID can be used to match the first 6 bytes of the UOWID in the CICS SMF 110 record.	display	8	144 090
ALTERNATE PCB DESTINATION LTERM LTERM name of the destination of an alternate PCB message inserted by this transaction, if any.	display	4	152 098

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 6 of 13)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED AREA	display	4	156 09C
SUMMARY DATA	**group**	**128**	**160 0A0**
NO TRANSACTIONS Number of transactions processed. The field is set to a value of 1 for a detail record and contains a count of the number of times this transaction was processed during the summary time span.	binary	4	160 0A0
INPUT Q TIME Amount of time, in hundredths of seconds, between arrival of the transaction in the input queue or BALG and the start time of processing. This field contains the sum of all queue wait times for the summary version record.	binary	4	164 0A4
ELAPSED TIME Amount of time, in hundredths of seconds, during which this transaction was actively being processed. This field contains the sum of all elapsed times for the summary version record.	binary	4	168 0A8
MESSAGE REGION PROGRAM CPU TIME Amount of CPU time, in thousandths of seconds, used by the application program in the message region to process this transaction. For DBCTL threads, the normal IMS message CPU time is replaced with the DRAW THREAD CREATE and processing time. This time is not the normal TCB time from CICS or ODBA. It is the TCB created to attach to the IMS control region and process the CICS or ODBA work. It does not include the control region CPU time. Since CICS does not report CPU time, there is no duplication of timings for CICS threads.	binary	8	172 0AC
MESSAGE DL1 CPU TIME Amount of message region CPU time, in thousandths of seconds, used by the DL/I analyzer to process database requests for this transaction.	binary	8	180 084
MESSAGE BUFFER CPU TIME Amount of message region CPU time, in thousandths of seconds, used by the IMS buffer handler during the processing of this transaction. If BHTO=OFF, this field is zero; CPU time is included in DL/I CPU.	binary	8	188 0BC
MESSAGE OPEN-CLOSE CPU TIME Amount of dependent region CPU time, in thousandths of seconds, used to open or close databases for this transaction. Only Fast Path DEDBs are opened in the dependent region.	binary	8	196 0C4
CONTROL DL1 CPU TIME Amount of control region CPU time, in thousandths of seconds, used by the DL/I analyzer to process the database requests for this transaction. If LSO=S, this field can contain CPU accumulated in DLISAS. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	204 0CC

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 7 of 13)

Element/Description	Format	Length	Offset DEC/HEX
CONTROL BUFFER CPU TIME Amount of control region CPU time, in thousandths of seconds, used by the buffer handler module during the processing of this transaction. If BHTO=OFF or CPU=DEP or NONE, this field is zero.	binary	8	212 0D4
CONTROL OPEN-CLOSE CPU TIME Amount of control region CPU time, in thousandths of seconds, used by the DL/I analyzer to open and close required database while processing requests issued for this transaction. If LSO=S, this field can contain CPU accumulated in DLISAS. For all Event Collector CPU data collection parameter options except ALL, this field is zero.	binary	8	220 0DC
USER AREA 2 Field that can be modified by the user.	display	4	228 0E4
NUMBER DB2 TRANSACTIONS Number of transactions that accessed DB2. This field is set to 1 for a detail record if any SQL calls were performed.	binary	4	232 0E8
ALLOCATED KSTORAGE MINUTES Number of KSTORAGE minutes, in thousandths of KSTORAGE minutes, consumed during transaction processing. KSTORAGE MINUTE=(ELAPSED TIME × STORAGE ALLOCATED)	binary	8	236 0EC
USED KSTORAGE MINUTES Number of KSTORAGE minutes, in thousandths of KSTORAGE minutes, used during transaction processing. KSTORAGE MINUTE=(ELAPSED TIME × STORAGE USED)	binary	8	244 0F4
STORAGE AVAILABLE For IMS, this field contains the amount of free storage available after IMS initialization and program preloading (if used). For DBCTL threads, the amount of storage available in the CICS or ODBA address space at the time the PSB is allocated.	binary	2	252 0FC
STORAGE USED Link-edited length of the program in 1K units. For DBCTL threads, this field is zero.	binary	2	254 0FE
DB2 SUBSYSTEM ID Four-character subsystem ID of DB2 system that the transaction accessed.	display	4	256 100
SYNC POINT RETURN CODE For a transaction that accessed Fast Path databases, this value is the result of SYNC point processing.	binary	2	260 104
BALANCING GROUP Q COUNT For a message-driven transaction, this value is the number of transactions queued on the same balancing group when this transaction went through SYNC point processing. For a summary IRUF, this field is summed for like transactions.	binary	2	262 106

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 8 of 13)

Element/Description	Format	Length	Offset DEC/HEX
NORMAL BUFFER ALLOCATION If this transaction ran in a region that supported Fast Path database access, this value is the NBA value from the JCL. For a summary IRUF, this field is summed for like transactions.	binary	2	264 108
OVERFLOW BUFFER ALLOCATION If this transaction ran in a region that supported Fast Path database access, this value is the OBA value from the JCL. For a summary IRUF, this field is summed for like transactions.	binary	2	266 10A
FAST PATH BUFFERS USED If this transaction accessed Fast Path database, this field shows how many Fast Path buffers were actually used. For a summary IRUF, this field is summed for like transactions.	binary	2	268 10C
FAST PATH CI CONTENTIONS Number of Fast Path CI contentions that occurred in the transaction SYNC interval. For a summary IRUF, this field is summed for like transactions.	binary	2	270 10E
FAST PATH WAIT FOR BUFFERS Number of times this transaction had to wait for a Fast Path buffer to become available. For a summary IRUF, this field is summed for like transactions.	binary	2	272 110
ROUTING CODE For a Fast Path message-driven region, this field is the routing code used to schedule this transaction.	display	8	274 112
FAST PATH FLAG Fast Path characteristics of this transaction if it accessed Fast Path database: W This transaction waited for the OBA latch. For a summary IRUF, this value is a counter from 0 to 9 that indicates the number of OBA latch waits.	display	1	282 11A
SPECIAL PROCESSING FLAG J Java message processing transaction (JMP) K Java batch message processing transaction (JBP) O DBCTL ODBA thread Q PWFI transaction (pseudo wait for input) W WFI transaction (wait for input)	display	1	283 11B
RESERVED AREA	display	4	284 11C
RESPONSE OPTION DATA	**group**	**68**	**288 120**
R MSGSW COUNT Number of message switches used to create this transaction (equals zero for transactions input from a terminal). Available only with the response option in effect.	binary	2	288 120

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 9 of 13)

Element/Description	Format	Length	Offset DEC/HEX
R MSGSENT COUNT Number of response messages sent to the originating terminal. For a detail IRUF, if R MSGSENT DATE > 0, this field is 1. For a summarized IRUF, field is summed for like transactions. Available only with the response option in effect. If this field overflows during summarization, the additional counts are kept in an overflow field.	binary	2	290 122
R MSGDEQ COUNT Number of response messages dequeued. For detail IRUF, if R MSGDEQ DATE > 0, this field is 1. For summarized IRUF, field is summed for like transactions. Available only with the response option in effect. If this field overflows during summarization, the additional counts are kept in an overflow field.	binary	2	292 124
R ORIGINAL DATE Julian date (yyyyddd) when this message switch was submitted (same as transaction arrival date). For MSC transactions, this value is the arrival time in the originating system. **Warning:** Clocks of the two systems cannot be synchronized.	binary	4	294 126
R ORIGINAL TIME Time of day, in hundredths of seconds, when this message switch was submitted (same as transaction arrival time). For MSC transactions, this value is the arrival time in the originating system. **Warning:** Clocks of the two systems cannot be synchronized.	binary	4	298 12A
R ARRIVAL DATE Julian date (yyyyddd) when this transaction arrived or the date of the message switch. Available only with the response option in effect and only in a detail record.	binary	4	302 12E
R ARRIVAL TIME Time of day, in hundredths of seconds, when this transaction arrived in the queue. For message switch transactions, it is the time the actual switched transaction was put in the message input queue; otherwise, this time is the transaction arrival time. Available only with the response option in effect and only in a detail record.	binary	4	306 132
R MSGISRT DATE Julian date (yyyyddd) when a response message was inserted into the queue. If no response was made, this field is zero. Available only with the response option in effect and only in a detail record.	binary	4	310 136
R MSGISRT TIME Time of day, in hundredths of seconds, when a response message was inserted into the queue. If no response was made, this field is zero. Available only with the response option in effect and only in a detail record.	binary	4	314 13A

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 10 of 13)

Element/Description	Format	Length	Offset DEC/HEX
R MSGSENT DATE Julian date (yyyyddd) when a response message was sent to the originating terminal. If no response was made, this field is zero. Available only with the response option in effect and only in a detail record.	binary	4	318 13E
R MSGSENT TIME Time of day, in hundredths of seconds, when a response message was sent to the originating terminal (first attempt to transmit). If no response was made, this field is zero. Available only with the response option in effect and only in a detail record.	binary	4	322 142
R MSGDEQ DATE Julian date (yyyyddd) a response message was dequeued. If no response was made, the field is zero. Available only in a detail record.	binary	4	326 146
R MSGDEQ TIME Time of day, in hundredths of seconds, when a response message was dequeued (after completion of transmission). If no response was made, this field is zero. Available only with the response option in effect and only in a detail record.	binary	4	330 14A
R INPUT LINE Input line number.	binary	2	334 14E
R INPUT PTERM Input physical terminal.	binary	2	336 150
R INPUT Q Amount of time, in hundredths of seconds, during which this transaction waited in the input queue for processing. It differs from INPUT Q TIME only in the case of message switches, where it is measured from the arrival of the message switch transaction on the queue. Available only with the response option in effect. This time is summarized if all records were created with this option.	binary	4	338 152
R OUTPUT Q Amount of time, in hundredths of seconds, during which this transaction waited in the output queue for processing. Output queue time is measured from R MSGISRT TIME to R MSGSENT TIME. Available only with the response option in effect. This time is summarized if all records were created with this option.	binary	4	342 156
R DE Q Amount of time, in hundredths of seconds, during which this transaction waited to be dequeued. Dequeue time is measured as R MSGDEQ TIME minus R MSGSENT TIME. Available only with the response option in effect. This time is summarized if all records were created with this option.	binary	4	346 15A
R RESPONSE Amount of time, in hundredths of seconds, from the transaction arrival time (of the original transaction) to the first attempt to transmit an output message to the originating terminal. If no response was made, this field is set to zero. Available only with the response option in effect. This time is summarized if all records were created with this option.	binary	4	350 15E

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 11 of 13)

Element/Description	Format	Length	Offset DEC/HEX
R MSGSW LEVEL Level of message switching to reach this transaction. A value of 1 indicates that the program processing the originating transaction from the terminal created this transaction as a message switch. A value of 2 indicates that another program was also invoked. Reported only in response reports and only with the response option in effect.	binary	2	354 162
RESERVED AREA	binary	6	356 164
BASIC RESOURCE CHARGES Charges for use of the basic resources.	**group**	**24**	**362 16A**
STORAGE ALOC CHG Charge for the number of ALLOCATED KSTORAGE MINUTES. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	362 16A
STORAGE USED CHG Charge for the number of USED KSTORAGE MINUTES used. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	366 16E
MSG PROG CPU CHG Charge for the number of message region CPU seconds used. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	370 172
MSG DL1 CPU CHG Charge for the number of message region DL/I CPU seconds used. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	374 176
CTL DL1 CPU CHG Charge for the number of control region DL/I CPU seconds used. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	378 17A
MINIMUM CHARGE Minimum transaction charge, expressed in dollars and tenths of a cent (999,999.999).	binary	4	382 17E
RESERVED AREA	display	4	386 182
TRANSACTION CHARGE AREA Total charges for the processing requirements for this transaction.	**group**	**20**	**390 186**
RESOURCE CHARGE Sum charge for all basic resource uses including storage, CPU times, database usage, and terminal activity. The charge is expressed in dollars and tenths of a cent (999,999.999).	binary	4	390 186
UNIT CHARGE Charge applied when the transaction has a special charged based on a charge per execution, as opposed to a charge based on resource usage. This field is zero if a special unit charge is not specified for this transaction.	binary	4	394 18A
PRORATE CHARGE Marginal increase or decrease in the base charge as a result of a discount or mark-up associated with a specified customer. This field is zero if a prorated charge is not specified for this customer's transactions.	binary	4	398 18E

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 12 of 13)

Element/Description	Format	Length	Offset DEC/HEX
BILLABLE CHARGE Final charge applied to this transaction. If this transaction is charged a per execution charge (for example, a unit charge of $.12 per execution), this field equals the sum of the UNIT CHARGE and PRORATE CHARGE fields. If this transaction is charged on the resource usage incurred, this field equals the sum of the RESOURCE CHARGE, MINIMUM CHARGE, and PRORATE CHARGE fields.	binary	4	402 192
CHARGE VARIANCE Difference between the RESOURCE CHARGE and UNIT CHARGE fields when the transaction is charged on a unit-charge basis. This field is zero if a unit charge option is not invoked.	binary	4	406 196
R MSGSENT COUNT OVERFLOW FLAG If Y, the halfword MSGSENT count at offset X'122' overflowed.	display	1	410 19A
R MSGSENT OVERFLOW COUNT Overflowed value. The total of MSGSENT count is the sum of the fullword counter and the halfword counter.	binary	4	411 19B
R MSGDEQ COUNT OVERFLOW FLAG If Y, the halfword MSGDEQ count at offset X'124' overflowed.	display	1	415 19F
R MSGDEQ OVERFLOW COUNT Overflowed value. The total of MSGDEQ count is the sum of the fullword counter and the halfword counter.	binary	4	416 1A0
DEPENDENT REGION DB2 CPU TIME Amount of dependent region CPU time, in thousandths of seconds, used by the transaction to process DB2 requests.	binary	8	420 1A4
DB2 CPU CHARGE Charge for the number of dependent region DB2 seconds used. The charge is in dollars and tenths of a cent (999,999.999).	binary	4	428 1AC
SAP EXIT USED Y SAP exit used N SAP exit not used	display	1	432 1B0
SHARED Q UOW (IMSID/STCK)	display	16	433 1B1
RESERVED AREA	display	1	449 1C1
NUMBER OF VARIABLE SEGMENTS Event Collector data collection parameter options set in IMFECP00 that are in effect during processing. Database: Usage and cost statistics for processing of this transaction accessing any DL/I database. DB2: Usage and cost statistics for processing of this transaction accessing any DB2 subsystem. Response: Response statistics quantifying the various response time groupings experienced in processing this transaction. (This segment type exists only in the summary version record.)	binary	2	450 1C2

Table B-1 Transaction Accounting Record Layout – Fixed Segment (Part 13 of 13)

Element/Description	Format	Length	Offset DEC/HEX
SMQ GROUP NAME Shared message queues group name.	display	8	452 1C4
RESERVED AREA	display	12	460 1CC
WORKLOAD MANAGER SERVICE CLASS Workload manager service class of the transaction.	display	8	472 1D8
PCB (TERMINAL) SEGMENT AREA Usage and cost statistics for the logical terminals used to submit this transaction. See “Record Format – Terminal Segment” on page B-15 for storage description.	display	192	480 1E0
FIXED RECORD DELIMITER This fixed record portion can be followed by 0 to 228 database segments.			672 2A0

Record Format – Terminal Segment

Table B-2 shows the record format for the terminal segment, which resides in the fixed portion of the transaction accounting record. The terminal segment contains usage and cost statistics for the logical terminals used to submit the transaction.

Table B-2 Transaction Accounting Record Layout – Terminal Segment (Part 1 of 4)

Element/Description	Format	Length	Offset DEC/HEX
TERMINAL SEGMENT One terminal segment for each terminal used as an originating LTERM in processing this transaction.	**group**	**192**	**000 000**
SEGMENT TYPE Code identifying this segment as a terminal segment is a value of P.	display	1	000 000
RESERVED AREA	display	1	001 001
TERMINAL NAME Name of the terminal associated with the statistics in this segment.	display	8	002 002
LINE NUMBER Physical line number of the terminal.	binary	4	010 00A
RESERVED AREA	display	2	014 00E
TERMINAL ACTIVITY	**group**	**096**	**016 010**

Table B-2 Transaction Accounting Record Layout – Terminal Segment (Part 2 of 4)

Element/Description	Format	Length	Offset DEC/HEX
CONNECT TIME Amount of time, in hundredths of seconds, that this terminal was connected to the system. This figure is not directly related to the transaction and only represents a means of storing the measured connect times for the various LTERMs.	binary	4	016 010
MESSAGE GET UNIQUE COUNT Number of times a MESSAGE GET UNIQUE was requested in processing this transaction.	binary	4	020 014
MESSAGE GET NEXT COUNT Number of times a MESSAGE GET NEXT was requested in processing this transaction.	binary	4	024 018
MESSAGE INSERT COUNT Number of times a MESSAGE INSERT was requested in processing this transaction.	binary	4	028 01C
MESSAGE PURGE COUNT Number of MESSAGE PURGEs issued by the program in processing this transaction.	binary	4	032 020
MESSAGE OTHER COUNT Number of other message calls including system service calls.	binary	4	036 024
INPUT CHARACTER COUNT Number of characters received as input over this LTERM in processing this transaction.	binary	8	040 028
INPUT SPA COUNT Number of characters input for this transaction from a scratch pad area in processing this transaction.	binary	8	048 030
INPUT OTHER COUNT Number of other input characters for this transaction.	binary	8	056 038
OUTPUT CHARACTER COUNT Number of characters sent as output to this LTERM in processing this transaction.	binary	8	064 040
OUTPUT SPA COUNT Number of characters output to a scratch pad area in processing this transaction.	binary	8	072 048
OUTPUT ALTTERM COUNT Number of characters output for this transaction to an LTERM other than the originating logical terminal.	binary	8	080 050
OUTPUT ALTTRAN COUNT Number of characters output to the message input queue as a result of a message switch by this transaction.	binary	8	088 058
OUTPUT OTHER COUNT Number of other output characters for this transaction.	binary	8	096 060

Table B-2 Transaction Accounting Record Layout – Terminal Segment (Part 3 of 4)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED AREA	binary	8	104 068
TERMINAL CHARGES Charges incurred for activity against this terminal in processing this transaction.	**group**	**60**	**112 070**
CONNECT CHG Charge for the number of hours of connect time for this terminal (see the definition for CONNECT TIME).	binary	4	112 070
MSG GU CHG Charge for the number of MESSAGE GET UNIQUEs issued.	binary	4	116 074
MSG GN CHG Charge for the number of MESSAGE GET NEXTs issued.	binary	4	120 078
MSG IN CHG Charge for the number of message INSERTs issued.	binary	4	124 07C
MSG PU CHG Charge for the number of message PURGEs issued.	binary	4	128 080
MSG OT CHG Charge for the number of other message calls.	binary	4	132 084
INPUT CHAR CHG Charge for the number of characters of input.	binary	4	136 088
INPUT SPA CHG Charge for the number of characters input for this transaction from a scratch pad area.	binary	4	140 08C
INPUT OTHER CHG Charge for the number of other input characters.	binary	4	144 090
OUTPUT CHAR CHG Charge for the number of characters of output.	binary	4	148 094
OUTPUT SPA CHG Charge for the number of characters output for this transaction to a scratch pad area.	binary	4	152 098
OUTPUT ALTTERM CHG Charge for the number of characters output to an LTERM other than the originating logical terminal.	binary	4	156 09C
OUTPUT ALTTRAN CHG Charge for the number of characters output to the message input queue as result of a message switch by this transaction.	binary	4	160 0A0
OUTPUT OTHER CHG Charge for the number of other output characters.	binary	4	164 0A4

Table B-2 Transaction Accounting Record Layout – Terminal Segment (Part 4 of 4)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED AREA	binary	4	168 0A8
RESERVED AREA	display	20	172 0AC

Record Format – DBD Variable Segment

Table B-3 shows the record format for the DBD variable segment of the transaction accounting record. DBD variable segments contain measurement data on database activity. There can be multiple occurrences of this segment, up to a maximum of 228. If there is one DB2 segment (see page B-21), the maximum number of DBD segments is 227.

Table B-3 Transaction Accounting Record Layout – DBD Variable Segment (Part 1 of 3)

Element/Description	Format	Length	Offset DEC/HEX
DBD SEGMENT One DBD segment for each database accessed in processing this transaction.	**group**	**132**	**000 000**
SEGMENT TYPE Code identifying this segment as a database segment is a value of D.	display	1	000 000
ORGANIZATION TYPE Code that identifies the type of organization used for this database: 1 HDAM 2 HIDAM 3 HISAM 4 HISAM-CASE2 5 HSAM 6 SSAM 7 Root index D DEDB J MSDB, nonrelated, key in segment A MSDB, nonrelated, LTERM key C MSDB, related, fixed G MSDB, related, dynamic H Partitioned HDAM I Partitioned HIDAM S Partitioned secondary index	display	1	001 001
DBD NAME Name of the database associated with the statistics in this segment. If the maximum number of segments is exceeded, an overflow segment is created with OTHERS in this field. If the Event Collector DBIO data collection parameter is IOWAITS, a special segment, ALLDBS, can be created to account for SYNC point write activity.	display	8	002 002

Table B-3 Transaction Accounting Record Layout – DBD Variable Segment (Part 2 of 3)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED AREA	display	2	010 00A
DMB POOL SIZE REQUIREMENT DMB buffer pool space, in bytes, required by this database.	binary	4	012 00C
DATABASE ACTIVITY	**group**	**56**	**016 010**
GET UNIQUE COUNT Number of times a GET UNIQUE was requested in processing this transaction.	binary	4	016 010
GET NEXT COUNT Number of times a GET NEXT was requested in processing this transaction.	binary	4	020 014
INSERT COUNT Number of times an INSERT was requested in processing this transaction.	binary	4	024 018
DELETE COUNT Number of times a DELETE was requested in processing this transaction.	binary	4	028 01C
REPLACE COUNT Number of times a REPLACE was requested in processing this transaction.	binary	4	032 020
OTHER COUNT Number of system service calls issued against this database by the transaction.	binary	4	036 024
KEY READS Number of reads issued to the key data area in satisfying the DL/I requests.	binary	4	040 028
KEY WRITES Number of writes issued to the key data area in satisfying the DL/I requests.	binary	4	044 02C
NONKEY READS Number of reads issued to the nonkey data area in satisfying the DL/I requests.	binary	4	048 030
NONKEY WRITES Number of writes issued to the nonkey data area in satisfying the DL/I requests.	binary	4	052 034
NO I Number of times that an input DL/I request did not require I/O because the requested data was found in the I/O buffer.	binary	4	056 038
NO O Number of times that an output DL/I request did not require I/O because the buffer write was charged to another user.	binary	4	060 03C
RESERVED AREA	binary	4	064 040
NUMBER OF REFERENCES Number of transactions that referenced this database.	binary	4	068 044

Table B-3 Transaction Accounting Record Layout – DBD Variable Segment (Part 3 of 3)

Element/Description	Format	Length	Offset DEC/HEX
DATABASE CHARGES Charges incurred for activity against this database in processing this transaction.	**group**	**52**	**072 048**
GU CHG Charge for the number of GET UNIQUEs issued.	binary	4	072 048
GN CHG Charge for the number of GET NEXTs issued.	binary	4	076 04C
IN CHG Charge for the number of INSERTs issued.	binary	4	080 050
DL CHG Charge for the number of DELETEs issued.	binary	4	084 054
RP CHG Charge for the number of REPLACEs issued.	binary	4	088 058
OTHER CHG Charge for the number of system service calls issued.	binary	4	092 05C
KEY READ CHG Charge for the number of key reads required.	binary	4	096 060
KEY WRITE CHG Charge for the number of key writes required.	binary	4	100 064
NONKEY READ CHG Charge for the number of nonkey reads required.	binary	4	104 068
NONKEY WRITE CHG Charge for the number of nonkey writes required.	binary	4	108 06C
NO I CHG Charge for the number of read requests that were satisfied by buffer hits and required no I/O action.	binary	4	112 070
NO O CHG Charge for the number of write requests that were satisfied by buffer hits and required no I/O action.	binary	4	116 074
DB2 DLI COUNT Number of DLI calls (GU + GN + ISRT + DLET + REPL) made if a transaction also accesses DB2.	binary	4	120 078
RESERVED AREA	display	8	124 07C

Record Format – DB2 Variable Segment (Optional)

Table B-4 shows the record format for the optional DB2 variable segment of the transaction accounting record. There is, at most, one occurrence of this segment. The segment will not be produced if the parameter FEATURE=NODB2 is specified in PARMLIB member IMFSYS00 (see PARMLIB member IMFSYSBB for more information).

Table B-4 Transaction Accounting Record Layout – DB2 Variable Segment (Part 1 of 2)

Element/Description	Format	Length	Offset DEC/HEX
DB2 SEGMENT One DB2 segment if DB2 was accessed by the transaction.	**group**	**132**	**000 000**
SEGMENT TYPE E Identifies this segment as a DB2 segment.	display	1	000 000
PLAN NAME DB2 plan name for this transaction.	display	8	001 001
RESERVED AREA	display	3	009 009
DB2 CALL ACTIVITY	**group**	**40**	**012 00C**
SELECT/FETCH CALLS Number of SQL SELECT and FETCH calls.	binary	4	012 00C
OPEN CALLS Number of SQL OPEN calls.	binary	4	016 010
INSERT CALLS Number of SQL INSERT calls.	binary	4	020 014
DELETE CALLS Number of SQL DELETE calls.	binary	4	024 018
UPDATE CALLS Number of SQL UPDATE calls.	binary	4	028 01C
DDL CALLS Number of SQL Data Definition Language calls made to DB2 (CREATE, DROP, ALTER, COMMENT and LABEL).	binary	4	032 020
DYNAMIC CALLS Number of SQL dynamic calls made to DB2 (PREPARE, DESCRIBE and EXECUTE).	binary	4	036 024
CONTROL CALLS Number of SQL control type calls made to DB2 (GRANT, REVOKE).	binary	4	040 028
OTHER CALLS Number of other SQL calls made to DB2 (EXPLAIN, LOCK, CLOSE).	binary	4	044 02C
NUMBER OF REFERENCES Number of transactions that referenced DB2.	binary	4	048 030

Table B-4 Transaction Accounting Record Layout – DB2 Variable Segment (Part 2 of 2)

Element/Description	Format	Length	Offset DEC/HEX
RESERVED AREA	display	4	052 034
DB2 CHARGES Charges incurred for activity against a DB2 subsystem. All charges are in dollars and tenths of a cent (999,999.999).	**group**	**24**	**056 038**
READ CHARGE Charge for the number of read-type calls issued (SELECT/FETCH, OPEN).	binary	4	056 038
UPDATE CHARGE Charge for the number of update-type calls issued (INSERT, DELETE and UPDATE).	binary	4	060 03C
DDL CHARGE Charge for the number of Data Definition Language calls issued.	binary	4	064 040
DYNAMIC CHARGE Charge for the number of dynamic SQL calls issued.	binary	4	068 044
CONTROL CHARGE Charge for the number of control-type calls issued.	binary	4	072 048
OTHER CHARGE Charge for the number of other DB2 calls issued.	binary	4	076 04C

Record Format – Response Variable Segment

Table B-5 shows the record format for the response variable segment of the transaction accounting record. This segment is produced during summarization. It contains response statistics quantifying the various response time groupings encountered in processing this transaction. This segment is produced only once in the summary record.

Table B-5 Transaction Accounting Record Layout – Response Variable Segment

Element/Description	Format	Length	Offset DEC/HEX
RESPONSE SEGMENT There is one segment per record in the summary version record that is a response type segment. The purpose of this segment is to maintain statistics on response time performance related to this transaction only.	**group**	**132**	**000 000**
SEGMENT TYPE Code identifying this segment as a response segment is a value of R.	display	1	000 000
RESERVED AREA	display	5	001 001
TRANSACTION CATEGORY Transaction category qualifying the response statistics maintained in this response segment, as specified in the IRUF summarization process	binary	2	006 006
WORK AREA	**group**	**8**	**008 008**
WORK RESPONSE TIME Work area.	binary	4	008 008
WORK RESPONSE COUNT Work area.	binary	4	012 00C
RESPONSE ENTRIES This group holds 12 statistical groups that quantify specified performance levels. Each entry consists of a response time threshold and a count. The response time is a performance objective specified as a parameter to the IRUF summarization process (for example, all transactions with response time less than 6.0 seconds). The response count is the number of transactions meeting the performance objective (for example, 150 executions of this transaction had response times less than 6.0 seconds but greater than the previous threshold).	**group**	**96**	**016 010**
RESPONSE TIME Response time specified for this performance group. The first response time is always set to zero. Any other entry having a response time of zero is an inactive group.	binary	4	016 010
RESPONSE COUNT Number of transactions having response times in excess of the response time specified in the next higher grouping.	binary	4	020 014
RESERVED AREA	display	20	112 070

Program Accounting Record Layout

This section describes the record format for the program accounting record (PAR). Table B-6 provides detailed descriptions of the record layout.

Table B-6 Program Accounting Record Layout (Part 1 of 6)

Element/Description	Format	Length	Offset DEC/HEX
RECORD IDENTIFICATION Code identifying record as a program accounting record (value P).	display	1	000 000
RESERVED AREA	display	1	001 001
RECORD VERSION MVIMS IRUF record version code. X'01' MVIMS version 3.2.00 or earlier X'02' MVIMS version 3.3.00 or later	display	2	002 002
PROGRAM IDENTIFICATION	**group**	**32**	**004 004**
RESERVED AREA	display	18	004 004
PROGRAM NAME Name of the program (PSB) for which the statistics in this record account.	display	8	022 016
APPC/IMS FLAG N non-APPC/IMS transaction Y APPC/IMS transaction O OTMA transaction	display	1	030 01E
SYSTEM IDENTIFICATION Identification code for the computing system on which this program was executed.	display	1	031 01F
IMS LEVEL IMS release level under which this program was executed.	display	4	032 020
NORMAL BUFFER ALLOCATION If this program was scheduled in a region that supported Fast Path database access, this value is the OBA value from the JCL. For a summary IRUF, this field is summed for all like programs.	binary	2	036 024
OVERFLOW BUFFER ALLOCATION If this program was scheduled in a region that supported Fast Path database access, this value is the OBA value from the JCL. For a summary IRUF, this field is summed for all like programs.	binary	2	038 026
FAST PATH BUFFER HIGH WATER MARK For all transactions processed during this scheduling, this value is the maximum number of Fast Path buffers used. For a summary IRUF, this field is summed for all like programs.	binary	2	040 028
RESERVED AREA	display	2	042 02A

Table B-6 Program Accounting Record Layout (Part 2 of 6)

Element/Description	Format	Length	Offset DEC/HEX
REGION ID Name assigned to the initiator used to schedule the program into a message region. For DBCTL threads, this field contains the CICS or ODBA address space jobname for the region name.	display	8	044 02C
CLASS Processing class of the transaction processed by this program.	display	3	052 034
RECORD TYPE Type of information stored in this record: D Detail level data S Summary level data	display	1	055 037
PROGRAM TYPE The identification for the program execution mode: (blank) Standard – message processing program (MPP) B Batch message processing program (BMP) C Conversational (MPP or JMP) D DBCTL CICS thread E CPI-C driven program (TPI) F Fast Path message-driven program (MDP) N Fast Path non-message-driven program (NDP) S Special record (MVIMS FINALPGM overhead record) U Fast Path utility (FPU)	display	1	056 038
ADDITIONAL PROGRAM TYPES The identification for the program execution mode: J Java message processing program (JMP) K Java batch message processing program (JBP) O DBCTL ODBA thread **Note:** A combination of a C program type and a J additional program type will result in an L program type (PSBTYPE) in a Formatted IRUF Record Report.	display	1	057 039
RESERVED AREA	display	2	058 03A
TIMESTAMPS (detail) Timestamps have two versions of this record: detail and summary. Both are specified in the RECORD TYPE field. (See subsequent description for summary version.) For a detail version record, the timestamps contain the dates and times for the basic time points of program start and end for one program execution. For purposes of sorting convenience, the start timestamp is stored both in the START STAMP and the RESERVED DATE1 stamp. The end timestamp is stored both in the END STAMP and the RESERVED DATE2 stamp.	n/a	n/a	n/a

Table B-6 Program Accounting Record Layout (Part 3 of 6)

Element/Description	Format	Length	Offset DEC/HEX
TIMESTAMPS (summarized) For a summary record, the timestamps reflect the range of time over which summarized data was collected for the number of programs executed (NO PROGRAMS). The start timestamp contains the earliest date and time encountered. The end timestamp contains the latest date and time encountered. The start timestamp is duplicated in RESERVED DATE1 and the end timestamp is duplicated in RESERVED DATE2.	n/a	n/a	n/a
PROGRAM START DATE Julian date (yyyyddd) when this program was started.	binary	4	060 03C
PROGRAM START TIME The time of day, in hundredths of seconds, when this program was started for execution.	binary	4	064 040
RESERVED DATE1 A duplicate version of the start timestamp is stored here.	group	8	068 044
RESERVED DATE2 A duplicate version of the end timestamp is stored here.	group	8	076 04C
PROGRAM END DATE Julian date (yyyyddd) this program was terminated.	binary	4	084 054
PROGRAM END TIME Time of day, in hundredths of seconds, when processing on this program was terminated.	binary	4	088 058
AGN Application group name assigned by the security system.	display	8	092 05C
RESERVED AREA	display	4	100 064
WORKLOAD MANAGER SERVICE CLASS Workload manager service class of the program.	display	8	104 068
RESERVED AREA	display	4	112 070
ADDRESS SPACE IDENTIFICATION ASID for the region in which this program executed.	binary	2	116 074
PERFORMANCE GROUP OS/390 SRM performance group for the region in which this program executed. If Goal Mode is active, this field contains a hexadecimal value of 0000.	binary	2	118 076
IMS SYSTEM IDENTIFICATION Identification code for the IMS system on which this transaction was processed (IMSID).	display	4	120 078
SMF SYSTEM ID ID code from SMCA for the computing system on which this program executed.	display	4	124 07C

Table B-6 Program Accounting Record Layout (Part 4 of 6)

Element/Description	Format	Length	Offset DEC/HEX
LSO FLAG LSO option in effect when this program was executed. It is one of the four IMS LSO options (N, Y, X, or S).	display	1	128 080
DB2 SUBSYSTEM ID Four-character subsystem ID of DB2 system that the transaction accessed.	display	4	129 081
RESERVED AREA	display	4	133 085
OS CODE Operating system (OS/390) under which this transaction was executed.	display	3	137 089
USER AREA Contents of the USER AREA contained in the MVIMS log records. This field can be used or modified by the user.	display	20	140 08C
NO PROGRAMS Number of times this program is scheduled.	binary	4	160 0A0
NO SYS ABENDS Number of system abends incurred in this program's execution.	binary	2	164 0A4
NO USER ABENDS Number of user abends incurred in this program's execution.	binary	2	166 0A6
ELAPSED TIME Amount of time, in hundredths of seconds, during which this program was actively in execution.	binary	4	168 0A8
PROG SCHD CPU TIME Amount of CPU time, in thousandths of seconds, used by this program in scheduling and terminating message processing. This time can include DB close activity.	binary	8	172 0AC
MSG REG OVHD CPU TIME The amount of overhead CPU time, in thousandths of seconds, used by this program. If LSO=S, this field can contain CPU time accumulated in DLISAS. For DBCTL threads, this field contains a zero.	binary	8	180 0B4
ALLOCATED KSTORAGE MINS Number of KSTORAGE minutes, in thousandths of seconds, consumed in the execution of this program. KSTORAGE MINUTE=(ELAPSED TIME × STORAGE ALLOCATED)	binary	8	188 0BC
STORAGE USED KSTORAGE MINS Number of KSTORAGE minutes, in thousandths of seconds, used in the execution of this program. KSTORAGE MINUTE=(ELAPSED TIME × STORAGE USED)	binary	8	196 0C4
MESSAGE GET UNIQUES Number of MESSAGE GET UNIQUEs issued by this program in processing the transactions.	binary	4	204 0CC

Table B-6 Program Accounting Record Layout (Part 5 of 6)

Element/Description	Format	Length	Offset DEC/HEX
MESSAGE GET NEXTS Number of MESSAGE GET NEXTs issued by this program in processing the transactions.	binary	4	208 0D0
MESSAGE INSERTS Number of MESSAGE INSERTs issued by the program in processing the transactions.	binary	4	212 0D4
MESSAGE PURGES Number of MESSAGE PURGEs issued by the program in processing the transactions.	binary	4	216 0D8
MESSAGE OTHERS Number of other message calls including system service calls.	binary	4	220 0DC
NUMBER PROGRAMS ACCESSING DB2 Number of schedulings of this program that accessed DB2. It is set to 1 for a detail record if any SQL calls were performed.	binary	4	224 0E0
RESERVED AREA	display	4	228 0E4
STORAGE ALLOCATED Largest contiguous area of storage in the dependent region available for the message processing program. STORAGE ALLOCATED is the amount of free storage available after IMS initialization and program preloading (if used). For DBCTL threads, this field contains the amount of storage available in the CICS or ODBA address space at the time the PSB is allocated.	binary	2	232 0E8
STORAGE USED Largest amount of storage this program actually used. STORAGE USED is the link-edited length of the program. For DBCTL threads, this field is zero.	binary	2	234 0EA
BUFFER POOL REQUIREMENT Highest level of PSB buffer pool space, in bytes, required by this program.	binary	4	236 0EC
DMB POOL REQUIREMENT Highest level of DMB buffer pool space, in bytes, required by this program.	binary	4	240 0F0
RESERVED AREA	display	15	244 0F4
COMPLETION STATUS	**group**	**5**	**259 103**
COMPLETION ID Code describing this program's termination status. The field is blank on a normal termination. S System abend U User abend	display	1	259 103
COMPLETION CODE Completion code (system or user) in EBCDIC format (for example, 0C7 or 4076). The completion-status field is only meaningful for detail version records.	display	4	260 104

Table B-6 Program Accounting Record Layout (Part 6 of 6)

Element/Description	Format	Length	Offset DEC/HEX
CONTROL PROGRAM DATA Segment contains data related to the processing of the control region and is not directly related to this program's measurement data. The control program data is maintained in this record only for processing convenience. The purpose of the data is to quantify the amount of CPU time used by the control region for the time period indicated.	**group**	**32**	**264 108**
IMS TIMESTAMP Start timestamp for the time period encompassing the measured control region CPU time.	**group**	**8**	**264 108**
CP START DATE Julian (yyyyddd) start date.	binary	4	264 108
CP START TIME Start time in hundredths of seconds.	binary	4	268 10C
IMS END STAMP End timestamp for the time period encompassing the measured control region CPU time.	**group**	**8**	**272 110**
CP END DATE Julian (yyyyddd) end date.	binary	4	272 110
CP END TIME End time in hundredths of seconds.	binary	4	276 114
CONT PROG CPU TIME Amount of CPU time, in thousandths of seconds, used by the control region for the time period defined in this section. (Includes attributable CPU from the TARs.)	binary	8	280 118
CTL REG OVHD CPU TIME Amount of nonattributable overhead CPU time, including DL/I SAS time, (in thousandths of seconds) used by the control region for the time period defined in this section.	binary	8	288 120
SAP EXIT CALLED Y SAP exit called N SAP exit not called	display	1	296 128
RESERVED AREA	display	33	297 129

Terminal Accounting Record Layout

Table B-7 describes the record format for the terminal (LTERM) accounting record (LAR). The table provides detailed descriptions of the record layout.

Table B-7 Terminal Accounting Record Layout (Part 1 of 3)

Element/Description	Format	Length	Offset DEC/HEX
RECORD IDENTIFICATION Code identifying record as a terminal accounting record (value L).	display	1	000 000
RESERVED AREA	display	1	001 001
RECORD VERSION MVIMS IRUF record version code. X'01' MVIMS version 3.2.00 or earlier X'02' MVIMS version 3.3.00 or later	display	2	002 002
TRANSACTION IDENTIFICATION	**group**	**32**	**004 004**
CUSTOMER ID User-specified code that correlates this terminal with its customer identity. This field is built by the user exit routine provided in the log file edit process.	display	18	004 004
HI VALUES This field is a reserved area, filled with a value of X'FFFFFFFFFFFFFFFF'.	display	9	022 016
SYSTEM IDENTIFICATION Identification code for the computing system to which this terminal was connected.	display	1	031 01F
IMS LEVEL IMS release level under which this transaction was processed.	display	4	032 020
LOGICAL TERMINAL NAME Name assigned to this logical terminal for this connect session.	display	8	036 024
RESERVED AREA	display	11	044 02C
RECORD TYPE Status of this record: D detail level data S summary level data	display	1	055 037
TERM TYPE Type of terminal connect in use: L leased line D dial-up line	display	1	056 038
RESERVED AREA	display	3	057 039
LOGON DATE Julian date (yyyyddd) when this logical terminal was logged on to the system.	binary	4	060 03C

Table B-7 Terminal Accounting Record Layout (Part 2 of 3)

Element/Description	Format	Length	Offset DEC/HEX
LOGON TIME Time of day, in hundredths of seconds, when this logical terminal was logged on to the system.	binary	4	064 040
RESERVED DATE1 A duplicate copy of the logon date and time is stored in this field.	group	8	068 044
RESERVED DATE2 A duplicate copy of the logoff date and time is stored in this field.	group	8	076 04C
LOGOFF DATE Julian date (yyyyddd) when this logical terminal was logged off of the system.	binary	4	084 054
LOGOFF TIME Time of day, in hundredths of seconds, when this logical terminal was logged off the system.	binary	4	088 058
USER ID User identification supplied by the security system (from the last transaction processed from this logical terminal).	display	8	092 05C
RESERVED AREA	display	37	100 064
OS CODE Operating system code under which this transaction was processed.	display	3	137 089
USER AREA Field that can be modified by the user.	display	20	140 08C
NO SESSIONS Number of terminal sessions encountered in the accounted span of time. For a leased terminal this figure represents the number of IMS sessions. For a dial-up terminal, it represents the number of actual dial-up connect sessions.	binary	4	160 0A0
TERMINAL CONNECT TIME Amount of time, in hundredths of seconds, during which this logical terminal was logged on to the system for this session.	binary	4	164 0A4
NUMBER OF TRANSACTIONS SUBMITTED Number of transactions submitted through this terminal during this logon session.	binary	4	168 0A8
NUMBER CHARACTERS INPUT Number of characters input from this terminal for message processing.	binary	8	172 0AC
NUMBER CHARACTERS OUTPUT Number of characters output to this terminal for message processing.	binary	8	180 0B4
NO CHARS OTHER IN Number of input characters transmitted by transactions submitted through this terminal that were directed to sources other than this LTERM.	binary	8	188 0BC

Table B-7 Terminal Accounting Record Layout (Part 3 of 3)

Element/Description	Format	Length	Offset DEC/HEX
NO CHARS OTHER OUT Number of output characters transmitted by transactions submitted through this terminal that were directed to sources other than this LTERM.	binary	8	196 0C4
MSG GU COUNT Number of MESSAGE GET UNIQUEs issued to this LTERM for this terminal session.	binary	4	204 0CC
MSG GN COUNT Number of MESSAGE GET NEXTs issued to this LTERM for this terminal session.	binary	4	208 0D0
MSG IN COUNT Number of message inserts issued by transactions from this LTERM for this terminal session. This count also includes message purges.	binary	4	212 0D4
MSG PURGE COUNT Number of message purges issued by transactions from this LTERM for this terminal session.	binary	4	216 0D8
MSG OTHER COUNT Number of other message calls (checkpoint, statistics, system service, and others) issued by transactions submitted from this LTERM.	binary	4	220 0DC
TOTAL RESPONSE Total of the queue times and elapsed times for all transactions processed. By dividing this value by the number of transactions, an average response time for this terminal can be derived. This time is expressed in seconds and hundredths of seconds.	binary	4	224 0E0
RESERVED AREA	display	100	228 0E4

Appendix C How Product Libraries Should Be Used

Several distributed libraries are included with your MAINVIEW products, including a parameter library (BBPARM), a sample library (BBSAMP), and a profile library (BBPROF). Use the contents of these distributed libraries as models to create site-customized product libraries, either manually or automatically, with AutoCustomization.

Warning! The distributed libraries should never be modified. If you change the distributed libraries, subsequent SMP maintenance will overwrite your changes.

Throughout the MAINVIEW documentation set, references to these libraries use the distributed name. However, when you need to make changes, be sure to use the corresponding library that has been customized for your site. Table C-1 lists the distributed name, the corresponding customized library created by AutoCustomization, and leaves space for you to note any other corresponding library that may have been created for your site.

Table C-1 Product Libraries

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

For more information about all the product libraries, see “Using MAINVIEW Product Libraries” in the *MAINVIEW Common Customization Guide* or “Using Product Libraries” in the *MAINVIEW Administration Guide*.

Appendix D Product Initialization Messages

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

Messages Changed with Version 3.3.10

Table D-1 lists the product initialization messages that were changed with version 3.3.10.

Table D-1 Product Initialization Messages Changed with Version 3.3.10

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

Messages Added with Version 3.3.00

Table D-2 lists the product initialization messages that were added with version 3.3.00.

Table D-2 Product Initialization Messages Added with Version 3.3.00

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

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

Messages Removed with Version 3.3.00

Table D-3 lists the product initialization messages that were removed with version 3.3.00, and it indicates the new messages that replaced them.

Table D-3 Product Initialization Messages Removed with Version 3.3.00

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

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

Messages Changed with Version 3.3.00

Table D-4 lists the MVIMS Online product initialization messages that were changed with version 3.3.00.

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

Table D-4 Product Initialization Messages Changed with MVIMS Online Version 3.3.00

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

Table D-5 lists the MVDBC product initialization messages that were changed with version 3.3.00.

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

Table D-5 Product Initialization Messages Changed with MVDBC Version 3.3.00

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

Index

A

B

C

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

W

END USER LICENSE AGREEMENT
NOTICE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

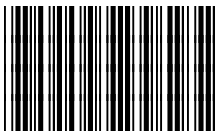
ARBITRATION. ANY DISPUTE BETWEEN YOU AND BMC ARISING OUT OF THIS AGREEMENT OR THE BREACH OR ALLEGED BREACH, SHALL BE DETERMINED BY BINDING ARBITRATION CONDUCTED IN ENGLISH. IF THE DISPUTE IS INITIATED IN THE AMERICAS REGION, THE ARBITRATION SHALL BE HELD IN NEW YORK, U.S.A., UNDER THE CURRENT COMMERCIAL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

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

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

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

Notes



22780