IBM Tivoli Decision Support for z/OS
Version 1.8.2

IMS CSQ Feature Guide and Reference

SH19-6825-13

IBM Tivoli Decision Support for z/OS
Version 1.8.2

IMS CSQ Feature Guide and Reference

SH19-6825-13

Note

Before using this information and the product it supports, read the information in “Notices” on page 167.

Thirteenth Edition (March 2015)

This edition applies to version 1, release 8, modification level 2 of Tivoli Decision Support for z/OS (program number 5698-B06) and to all subsequent releases and modifications until otherwise indicated in new editions.

© COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION. ALL RIGHTS RESERVED. Note to U.S. Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp. 1994, 2015

Contents

Figures

Tables

Preface

This book provides information about the IMS CSQ feature of IBM® Tivoli® Decision Support for z/OS® (hereafter referred to as Tivoli Decision Support for z/OS).

This book allows you to install and use the IMS CSQ feature of Tivoli Decision Support for z/OS. It describes:

- How to collect and report performance data generated by Information Management System (IMS)
- Performance characteristics shown in Tivoli Decision Support for z/OS reports, so you can analyze the characteristics of your system

The terms MVS™, OS/390®, and z/OS are used interchangeably throughout this book.

Who should read this book

IMS CSQ feature Guide and Reference is for:

- Anyone who analyzes IMS performance
- Anyone responsible for establishing or meeting enterprise-wide service-level objectives for IMS user groups
- Anyone who designs, monitors, or tunes IMS or the databases it uses
- Tivoli Decision Support for z/OS administrators (primarily as a reference to table and report definitions)
- Users with various backgrounds who are interested in analyzing IMS performance data and improving IMS performance

Use this book for guidance in collecting IMS-generated performance data and generating the reports shipped with the IMS Performance feature. This book explains how to use Tivoli Decision Support for z/OS reports to both understand and evaluate the performance of your systems. It helps you identify any problems indicated by your data and offers suggestions about how you can monitor, analyze, and improve IMS performance.

Publications

This section lists publications in the Tivoli Decision Support for z/OS library and any other related documents. It also describes how to access Tivoli publications online, how to order Tivoli publications, and how to submit comments on Tivoli publications.

Tivoli Decision Support for z/OS library

The following documents are available in the Tivoli Decision Support for z/OS library:

- *Administration Guide and Reference*, SH19-6816

  Provides information about initializing the Tivoli Decision Support for z/OS database and customizing and administering Tivoli Decision Support for z/OS.
- *AS/400 System Performance Feature Guide and Reference*, SH19-4019

Provides information for administrators and users about collecting and reporting performance data generated by AS/400 systems.

- *CICS Performance Feature Guide and Reference*, SH19-6820

  Provides information for administrators and users about collecting and reporting performance data generated by Customer Information and Control System (CICS®).
- *Distributed Systems Performance Feature Guide and Reference*, SH19-4018

  Provides information for administrators and users about collecting and reporting performance data generated by operating systems and applications running on a workstation.
- *Guide to Reporting*, SH19-6842

  Provides information for users who display existing reports, for users who create and modify reports, and for administrators who control reporting dialog default functions and capabilities.
- *IMS Performance Feature Guide and Reference*, SH19-6825

  Provides information for administrators and users about collecting and reporting performance data generated by Information Management System (IMS).
- *Language Guide and Reference*, SH19-6817

  Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.
- *Messages and Problem Determination*, SH19-6902

  Provides information to help operators and system programmers understand, interpret, and respond to Tivoli Decision Support for z/OS messages and codes.
- *Network Performance Feature Installation and Administration*, SH19-6901

  Provides information for network analysts or programmers who are responsible for setting up the network reporting environment.
- *Network Performance Feature Reference*, SH19-6822

  Provides reference information for network analysts or programmers who use the Network Performance feature.
- *Network Performance Feature Reports*, SH19-6821

  Provides information for network analysts or programmers who use the Network Performance feature reports.
- *Resource Accounting for z/OS*, SH19-4495

  Provides information for users who want to use Tivoli Decision Support for z/OS to collect and report performance data generated by Resource Accounting for z/OS.
- *Resource Accounting*, SH19-6818

  Provides information for performance analysts and system programmers who are responsible for meeting the service-level objectives established in your organization.
- *System Performance Feature Guide*, SH19-6819

  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM®, zLinux, and their subsystems, performance data.
- *System Performance Feature Reference Volume I*, SH19-4494

  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for z/OS to analyze z/OS, z/VM, zLinux, and their subsystems, performance data.
- *System Performance Feature Reference Volume II*, SC23-7966

Provides information about the functions and features of the Usage and Accounting Collector.

- *IBM Online Library z/OS Software Products Collection Kit*, SK3T-4270

  CD containing all z/OS documentation.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

http://www.ibm.com/ibm/terminology

Accessing publications online

IBM posts publications for this and all other Tivoli products, as they become available and whenever they are updated, to the Tivoli software information center Web site. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

http://www.ibm.com/software/tivoli/library/

Scroll down and click the **Product manuals** link. In the Tivoli Technical Product Documents Alphabetical Listing window, click the Tivoli Decision Support for z/OS link to access the product library at the Tivoli software information center.

Note: If you print PDF documents on other than letter-sized paper, set the option in the **File ″ Print** window that allows Adobe Reader to print letter-sized pages on your local paper.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, see the Accessibility Appendix in the *Administration Guide and Reference*.

Tivoli technical training

For Tivoli technical training information, refer to the following IBM Tivoli Education Web site:

http://www.ibm.com/software/tivoli/education/

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.

- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Software Support.

For more information about these three ways of resolving problems, see Appendix F, "Support information," on page 163.

Conventions used in this book

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

The following terms are used interchangeably throughout this book:
- MVS, OS/390, and z/OS.
- VM and z/VM.

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)
- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

`Monospace`

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Changes in this edition

This edition is an update of the previous edition of the same book. The changes relate to 1.8.2 GA APAR documentation.

Part 2. IMS CSQ feature reference

Chapter 6. Data tables and lookup tables

- Table 8 on page 68. New components "Transaction Level Statistics" and "Key Performance Metrics" added.
- “KPM_IMS_TRAN_H,_D, _W” on page 98. New table relating the Key Performance Metrics component.

Part 3. Appendixes

Appendix A. Reports

- “Key Performance Metrics IMS reports” on page 145. New Key Performance Metrics reports:
  - “KPM IMS Processing Times by Transaction by Hour report” on page 145.
  - “KPM IMS Processing Times by PSB Name by Hour report” on page 146.
  - “KPM IMS Processing Times by Region Type by Hour report” on page 147.
  - “KPM IMS Average Enqueues by Transaction by Hour report” on page 148.
  - “KPM IMS DB Activity by Region by Program Name by Hour report” on page 149.

Except for editorial changes, updates to this edition are marked with a vertical bar [|] to the left of the change.

Changes in this edition

Part 1. IMS CSQ Feature Guide

Chapter 1. Introducing the IMS CSQ feature

Tivoli Decision Support for z/OS is a reporting system that collects performance data logged by computer systems, summarizes the data, and presents it in a variety of forms for use in systems management. Tivoli Decision Support for z/OS consists of a base product and several optional features.

The base product includes:

- Interactive System Productivity Facility (ISPF) host reporting and administration dialogs
- The Tivoli Decision Support for z/OS log collector program
- Log and record definitions for all records used by the product features

Each feature provides:

- Update definitions for DB2® tables
- Table definitions
- Report definitions

Tivoli Decision Support for z/OS enables you to collect large volumes of data and keep the space to store it at acceptable levels. The database stores all reporting data, which comes from several sources. For example, logs from System Management Facilities (SMF), Resource Management Facility (RMF™), Customer Information and Control System (CICS), and Information Management System (IMS) can be consolidated into a single report. If you install all components of all Tivoli Decision Support for z/OS features and set system and subsystem data-recording parameters as recommended for each feature, you can ensure a steady supply of data about the operation of your entire computer center.

The IMS CSQ feature supports data from systems running IMS version 12 (C), release 1, and later. You use the pre-merged or internally-merged IMS system log data set (SLDS) to generate data for the predefined tables and reports in the IMS CSQ feature.

Preparing IMS log data for collection

Before you collect IMS log data into Tivoli Decision Support for z/OS DB2 tables, you can preprocess the log data in various ways (shown in Figure 1 on page 4):

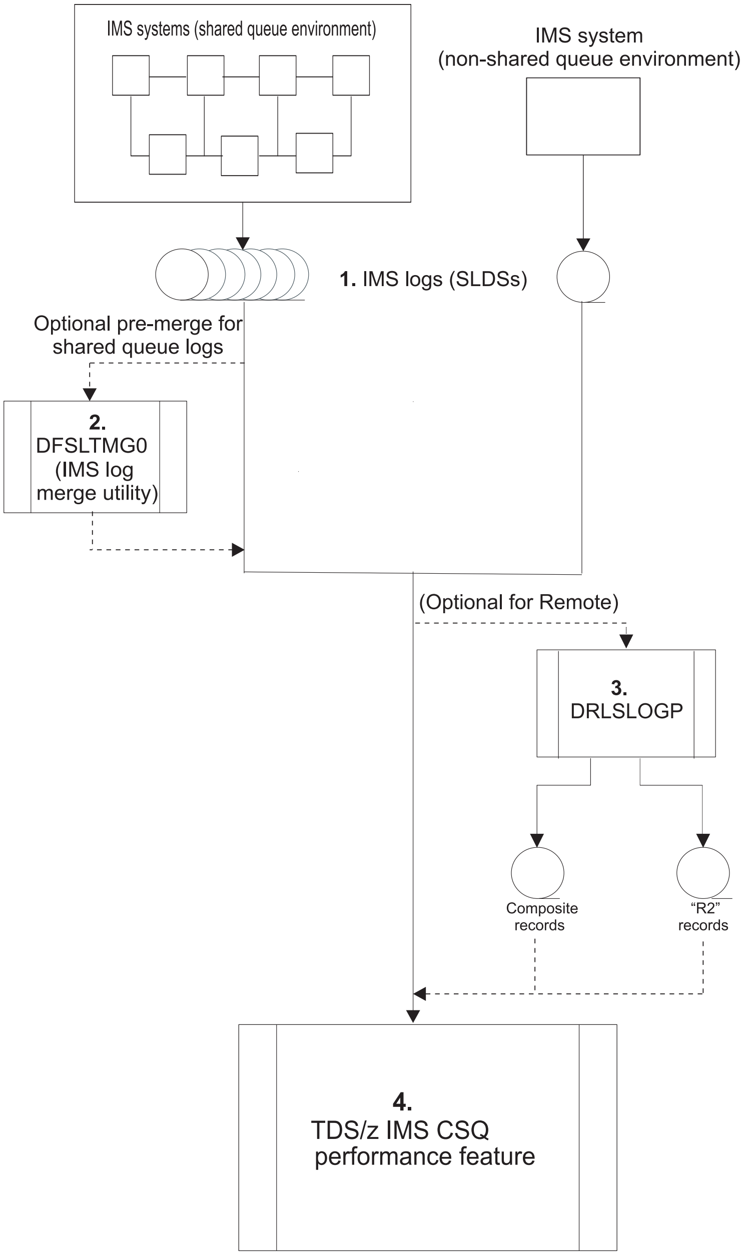


Figure 1. Preparing IMS log data for collection

The following list relates to Figure 1:

1. SLDS
Every IMS system produces a system log data set (SLDS) during the IMS archive process. The IMS systems that produce these logs could be members of an IMS shared group, or could be stand-alone IMS systems. Furthermore, these IMS systems could be local or remote to the z/OS system on which Tivoli Decision Support for z/OS processing occurs.

2. DFSLTMG0
Logs produced by IMS systems in a shared group must be merged for collection. You can use the IMS utility DFSLTMG0 to merge the logs prior to collection, or you can use the Tivoli Decision Support for z/OS internal merge feature to merge the logs as part of the collection process (see "Collecting IMS log data" on page 5 for more information).

3. DRLSLOGP

DRLSLOGP is a stand-alone batch program that calls the log and record procedures. It does not update the DB2 tables. Instead, it produces composite records or R2 records as output.

DRLSLOGP is usually used to obtain a log (either DRLICOMP which is a composite record log, or DRLIRPT2 which is an R2 record log) smaller than the original IMS log. You can use it as an input log in a subsequent collect to populate the product tables.

Logs produced on a system that is remote to the z/OS system on which Tivoli Decision Support for z/OS processing will occur must be transported or transmitted to the collection site. DRLSLOGP can reduce the amount of data that needs to be sent. It gives you the choice of creating either composite records containing all the information required by Tivoli Decision Support for z/OS for IMS data collection, or R2 records containing transaction-oriented data only.

For more information go to the relevant topics.

- Data flow diagrams for the log collector and DRLSLOGP, see Chapter 3, “Understanding data flow through the IMS CSQ feature,” on page 23.
- Running the log collector and DRLSLOGP, see Chapter 4, “Administering the IMS CSQ feature,” on page 29.
- Collecting R2 records, see “Using the IMS light feature” on page 42.

4. IMS CSQ feature

The IMS CSQ feature can process the following:

- Raw IMS logs from shared groups or from stand-alone IMS systems.
- IMS logs merged by the IMS log merge utility DFSLTMG0.
- Composite, R2, and Transaction Level Statistics records produced at a remote site by the Tivoli Decision Support for z/OS utility DRLSLOGP.

Collecting IMS log data

The process of collecting IMS performance data into DB2 tables is called a *collect*. Figure 2 on page 6 shows an overview of this process.

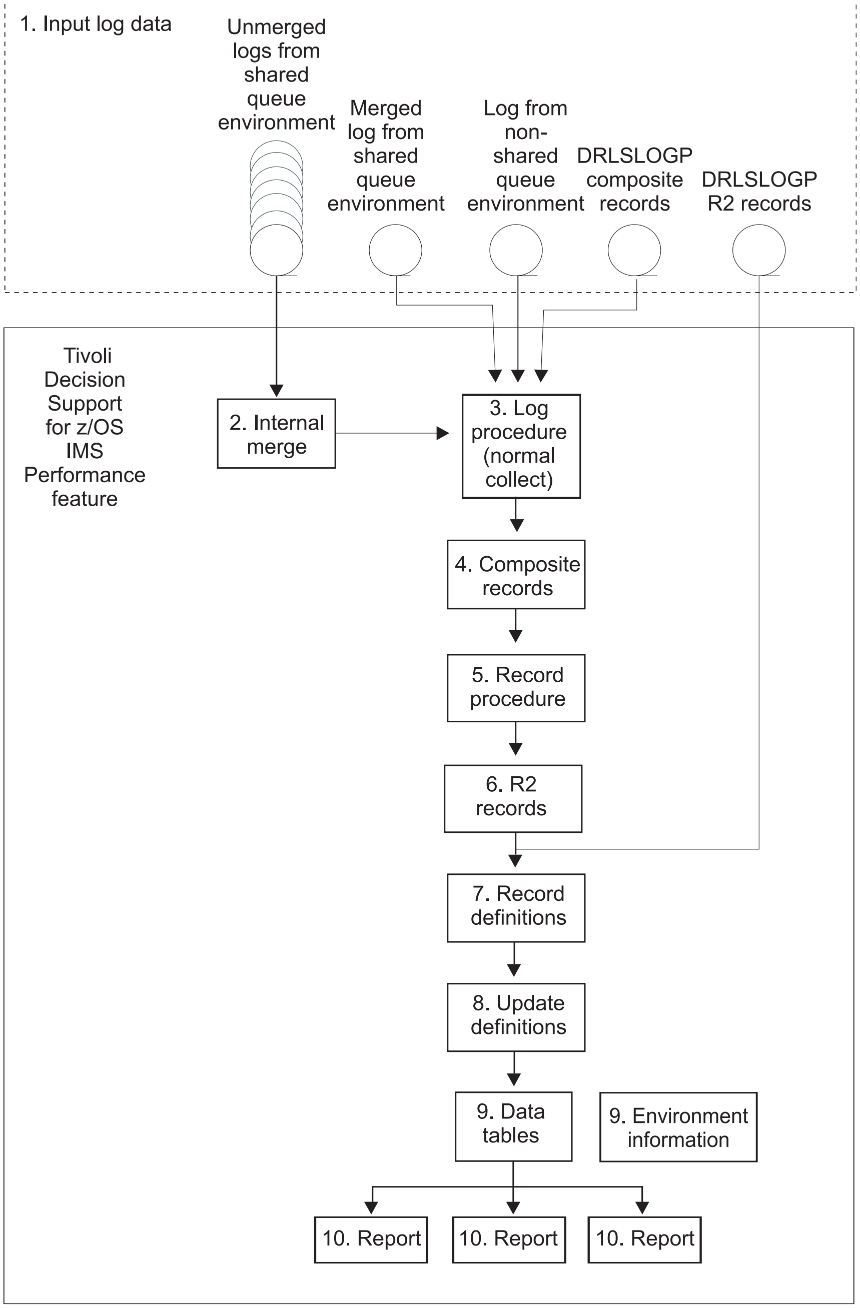

1. Input log data
Unmerged logs from shared queue environment
Merged log from shared queue environment
Log from non-shared queue environment
DRLSLOGP composite records
DRLSLOGP R2 records
Tivoli Decision Support for z/OS IMS Performance feature
2. Internal merge
3. Log procedure (normal collect)
4. Composite records
5. Record procedure
6. R2 records
7. Record definitions
8. Update definitions
9. Data tables
9. Environment information
10. Report
10. Report
10. Report

Figure 2. Overview of data collection using the IMS CSQ feature

The process is as follows:

1. IMS CSQ feature input log data

The Tivoli Decision Support for z/OS IMS CSQ feature can process the following:

- Raw IMS logs from shared groups or from stand-alone IMS systems.
- IMS logs merged by the IMS log merge utility DFSLTMG0.
- | Composite, R2, and Transaction Level Statistics records produced at a
  | remote site by the Tivoli Decision Support for z/OS utility DRLSLOGP.

All of these possible input sources are presented to Tivoli Decision Support for z/OS as input log data.

2. Internal merge
If the input data to the Tivoli Decision Support for z/OS collect consists of logs from IMS systems in a shared group, and you did not preprocess the logs with the IMS merge utility DFSLTMG0, then you must use the internal merge feature to merge the logs at the time of collection. Activate the internal merge by specifying SQNLOGS as an input parameter. For more information about this, see “Running the log collector” on page 31.

3. IMS CSQ feature log definitions
The IMS CSQ feature provides a log definition for collecting raw IMS records using a log procedure. The log procedure extracts information from selected log records related to the same event, such as the processing of a set of related transactions, and produces composite records. Composite records produced by DRLSLOGP are also processed by the log procedure.

4. Composite records
When the log procedure processes IMS log records, it produces a composite record for each completed event. When the log procedure processes composite records produced by DRLSLOGP it simply passes them on for further processing.

5. Record procedure
Composite records containing transaction-specific information are processed by a record procedure. The record procedure breaks the information in the composite record up into one R2 record for each transaction.

6. R2 records
Each R2 record contains data for the processing of a single IMS transaction. For a normal collect the R2 records are created from the composite records produced by the log procedure as part of the collection process. For a 'light' collect the R2 records produced by DRLSLOGP are used as input.

7. Record definitions
Composite records produced by the log procedure and not further processed by the record procedure are described by record definitions provided by Tivoli Decision Support for z/OS. R2 records produced by the record procedure are also described by record definitions provided by Tivoli Decision Support for z/OS.

8. Update definitions
Update definitions reference fields in the composite and R2 records using their related record definitions, and update Tivoli Decision Support for z/OS data tables.

9. Data tables and environmental information
The IMS CSQ feature uses the composite and R2 records, along with user-supplied data in lookup tables, to update the data tables. User-supplied data consists of IMS application names, and period and shift descriptions. The IMS performance data is stored in a series of data tables that are used when processing data and creating reports.

10. Reports
Tivoli Decision Support for z/OS creates reports from the information stored in the data tables. In addition to the reports provided with the IMS CSQ feature, you can create your own reports using, for example, the Query Management Facility (QMF) prompted query language.

Selecting Tivoli Decision Support for z/OS IMS CSQ feature components

The IMS Performance feature components are:

- IMS 11.1 CSQ collect component
- IMS 12.1 CSQ collect component
- IMS 13.1 CSQ collect component
- Key Performance Metrics (KPM) - IMS collect component

In addition to the components specific to CSQ processing, the following collect components are available for further analysis of IMS log data .

- IMS 11.1 log records component
- IMS 12.1 log records component
- IMS 13.1 log records component

The collect components

The collect components are divided into subcomponents. Each subcomponent collects data into DB2 tables and includes predefined reports. The subcomponents are:

Transaction Transit Time

This subcomponent collects information about transactions and BMPs. Information available includes system response times, system transaction volumes, CPU and database utilization, and transaction detail.

System Tran Transit Time

This subcomponent collects information about transactions and response times summarized by the IMS system.

Extended System Transaction Transit Time

This subcomponent collects information about transactions and performance. It is an extension of the System Tran Transit Time subcomponent with the addition of monthly aggregation. Information available includes system response times, system transaction volumes, CPU and database utilization, and transaction detail.

Accounting subcomponent

This subcomponent collects information about resource consumption of IMS systems, regions, and applications.

Availability subcomponent

This subcomponent collects information about availability data of IMS systems, regions, and applications.

Extended Accounting

This subcomponent collects information about resource consumption with the addition of monthly aggregation. It is an extension of the Account part of the Account and Availability subcomponent (IMS_PSB_ACCOUNT_*x*).

Statistics

This subcomponent records statistical information about buffer and pool usage.

HALDB OLR

This subcomponent collects information about the High Availability Large Database Online Reorganization (HALDB OLR) process.

Choosing between standard and extended subcomponents

The main differences between the new extended subcomponents and the standard subcomponents are as follows:

Note: You cannot use both standard and extended subcomponents.

Table 1. Differences between standard and extended subcomponents

Standard	Extended
The IMS_PSB_ACCOUNT_x table contains hourly and daily statistics only.	The IMS_PSB_ACCOUNT2_x table contains monthly statistics and additional key fields, PERIOD_NAME (name of the period or shift in which the activity occurred) and MVS_SYSTEM_ID.
The IMS_SYSTEM_TRAN_x tables do not contain the following columns: • MVS_SYSTEM_ID • PROGRAM_NAME • TRANSACTION_CLASS • PERFORMANCE_GROUP	Key fields have been reordered in the IMS_SYSTEM_TRAN2_x tables and the following columns have been added to the key: • MVS_SYSTEM_ID • PROGRAM_NAME • TRANSACTION_CLASS • PERFORMANCE_GROUP
The IMS_SYSTEM_TRAN_x tables do not contain the PGM_CPU_APPROX column.	The IMS_SYSTEM_TRAN2_x tables contain the PGM_CPU_APPROX column. This column represents the sum of the approximate number of CPU seconds of program execution time while transactions are active.

For a description of these tables and columns, see Chapter 6, “Data tables and lookup tables,” on page 67.

Availability for IMS resources

Availability for IMS resources relates to the following types of resource:
- IMS subsystem
- IMS region

IMS availability is obtained by looking at the subsystem availability and at the real usage of the applications.

For the IMS subsystem, the following IMS record types are used to track availability:
- Record type X'06': IMS subsystem Start/Stop
- Record type X'4001': IMS system checkpoint

For the IMS region, the following IMS record type is used to track availability:
- Record type X'47': IMS region checkpoint

The following tables are used:
- IMS_AVAIL_RESOURCE
- IMS_AVAILABILITY_T,_D,_W

Availability for IMS resources

The IMS_AVAIL_RESOURCE lookup table defines which IMS resources are used for tracking availability. These values are used during the update of the IMS_AVAILABILITY_T table. It also contains the schedule names and availability objectives to use for the different resources. These values are used in the IMS_AVAILABILITY_D and _W tables.

Updating other lookup and control tables

About this task

The IMS CSQ feature uses the DAY_OF_WEEK and PERIOD_PLAN control tables, which are also used by other Tivoli Decision Support for z/OS features. Check these tables and update them as needed.

For information about these tables, see the *Administration Guide*.

If you have installed the z/OS System (MVS) component of the Tivoli Decision Support for z/OS System Performance feature, you can use it to collect and report on IMS region activity. This information can be helpful when you need reports on IMS availability. To obtain this data, you need to update the MVS_WORKLOAD_TYPE table. For information, see Tivoli Decision Support for z/OS System Performance Feature Reference Volume I.

You can use collect parameters to set the values for the following data table columns:

```
IMS_SYSTEM_ID   - IMS system ID
SYSPLEX_NAME    - sysplex name
MVS_SYSTEM_ID   - MVS system ID
IMS_CTRL_REGION - IMS control region
IMS_APPLID      - VTAM application ID name
```

However, the specification of only a single value for each of these parameters is allowed, and therefore it is not advisable to specify these parameters when you are collecting data from multiple IMS systems in a shared queue environment. In this case you should populate lookup table IMS_SYSTEM_NAMES with your IMS system IDs.

The following steps describe how TDSz populates column IMS_SYSTEM_ID during the collect:

- TDSz derives IMS_SYSTEM_ID from the IMS records.
- When IMS_SYSTEM_ID cannot be derived from the IMS records, column IMS_SYSTEM_ID is populated based on the following conditions:
  - When IMS_SYSTEM_ID is specified with the SET statement, IMS_SYSTEM_ID is used to populate column IMS_SYSTEM_ID.
  - When IMS_SYSTEM_ID is not specified with the SET statement, column IMS_SYSTEM_ID is set to '$UNKNOWN'.

The following steps describe how TDSz populates columns SYSPLEX_NAME, MVS_SYSTEM_ID, IMS_CTRL_REGION, and IMS_APPLID during the collect:

- TDSz uses lookup table IMS_SYSTEM_NAMES to obtain values for these columns.
- When values are not available for these columns in lookup table IMS_SYSTEM_NAMES, these columns are populated based on the following conditions:
  - When values are specified with the SET statement, these values are used to populate these columns.

- When values are not specified with the SET statement, the columns are set to '$UNKNOWN'.

Lookup table IMS_SYSTEM_NAMES contains no default values.

If you are collecting data from multiple systems, IMS_SYSTEM_NAMES must contain information for all of these systems.

The following sample shows how you can populate the lookup table:

```
IMS_SYSTEM_ID SYSPLEX_NAME MVS_SYSTEM_ID IMS_CNTL_REGION IMS_APPLID
%             ?            ?             ?               ?
IMSA          MYPLEX       MVS1          REGIONA         IMS1
IMSB          MYPLEX       MVS2          REGIONB         IMS2
IMSC          MYPLEX       MVS3          REGIONC         IMS3
```

For more instructions about using the administration dialog to edit the contents of this lookup table, see the chapter that explains working with tables and update definitions in the *Administration Guide and Reference*.

Using the IMS CSQ feature

About this task

Before using the IMS CSQ feature, verify that Tivoli Decision Support for z/OS is collecting the right data, storing the data correctly, and using the proper data to generate the reports. Verify also that the IMS_AVAIL_RESOURCE lookup table contains the appropriate definitions. For information about verifying your installation, see the *Administration Guide*.

For information about administering Tivoli Decision Support for z/OS, see "Part 3. Administering Tivoli Decision Support for z/OS" in the *Administration Guide and Reference*. For specific information about running the IMS CSQ feature, see Chapter 4, “Administering the IMS CSQ feature,” on page 29.

Chapter 2. How the IMS CSQ feature uses log and record procedures

This topic explains the use of log and record procedures within the IMS CSQ feature.

The IMS CSQ feature log procedure takes one or more records from a log and creates another record called a *composite record* that includes data from the input records. The log procedure defines the fields taken from each input record and the contents of the output record.

The IMS CSQ feature record procedure processes composite records created by the log procedure and prepares them for collection using record and update definitions.

The log procedure

The IMS CSQ feature is based on a log processing routine, that is, a log procedure designed to process selected records from all the IMS systems that are members of the same IMS shared group. The procedure produces composite records at IMS transaction level (full-function or fast-path), as well as composite records containing statistical and accounting information. The log procedure copies only the required IMS log record fields to the composite records.

The information related to a transaction is distributed among numerous log record types. The log procedure associates information related to a transaction using the IMS shared-queue *unit of work* (UOW) key found in many log records. This key is unique through all the IMS logs produced by the systems in an IMS shared group. The log procedure also uses the IMS log record*recovery token* (RTKN) as a key to associate related log records.

Unit of work and recovery token keys

Since IMS version 6, log records related to transaction processing have contained a unit of work identifier (or key) to uniquely identify messages processed on the IMS message queues. The unit of work identifier is a 34-byte field found in record types X'01', X'03', X'31', X'33', X'35', X'36', X'37', X'38', X'5901', X'5903', X'5937', and X'5938' with the following format:

Originating system message identifier (bytes 1-16)
The message identifier assigned by the originating IMS system, comprising the IMS system identifier and a unique token obtained from the system clock. This is referred to by Tivoli Decision Support as the *originating unit of work* (OUOW).

Processing system message identifier (bytes 17-32)
The message identifier assigned by the processing IMS system, comprising the IMS system identifier and a unique token obtained from the system clock. This is referred to by Tivoli Decision Support as the *processing unit of work* (PUOW).

Flag bytes (bytes 33-34)
Not used by Tivoli Decision Support.

The OUOW does not change during the life of the transaction, and is therefore common to all log records containing a unit of work identifier that are related to a transaction. The PUOW can be different for some of the records, depending on which IMS system in a shared group processed the message, and the role of the log record in relation to the transaction.

The IMS log record recovery token represents work performed by a program during a commit interval. The recovery token is a 16-byte field found in record types X'07', X'08', X'31', X'35', X'56FA', X'5901', X'5903', X'5937', and X'5938' with the following format:

IMS system identifier
8 bytes.

Schedule count
4 bytes.

Commit count
4 bytes (zero when the program is scheduled and incremented every time the program commits).

The first 12 bytes of the recovery token do not change as long as the program remains scheduled, and are therefore common to all log records containing a recovery token that are related to the program.

Record grouping

The IMS log records containing transaction-level information are grouped together by the log procedure using the unit of work and recovery token keys. The groupings are collections of related records that represent database and data communication activity taking place for the transaction. The groupings are organized as follows:

OUOW group
This is the primary grouping. All log records containing a UOW identifier related to a primary transaction will have the same OUOW.

PUOW group
All log records containing a unit of work identifier related to the same message for a transaction will have the same PUOW. All PUOW groups sharing the same OUOW will be in the same primary grouping.

PSB group
All log records related to program scheduling that contain a recovery token will have the same 12-byte RTKN. Some log records in PUOW groups (for example the type X'31' of an input PUOW group) also contain recovery tokens, which are used to relate the PSB group (program) information to the PUOW group (transaction) information.

Composite records and subtypes

The log procedure produces composite records with a record type controlled by the **RECTYPE** log procedure parameter, which has a default value of X'FF'. Each composite record has a subtype that determines how the record is subsequently processed.

Composite records with subtype X'01' contain transaction-level information and are processed by the IMS CSQ feature record procedure. The log procedure can write composite records for incomplete transactions when the **TABLEFLUSH** or **WRITEPENDING** log procedure parameters are specified.

The remaining composite records have subtypes derived from the IMS log record type on which the composite record is based. These composite records are processed directly by Tivoli Decision Support for z/OS using record and update definitions.

Handling of special IMS cases within the IMS CSQ

The IMS CSQ feature can handle the following cases:

Multiple segment input
The log record for the first (or only) segment processed creates input OUOW and PUOW groups. The log procedure skips log records for subsequent segments.

Single segment input with operator logical Paging output
All the type X'31' log record output segments are linked to the output PUOW group. The log procedure skips subsequent X'31' segments and does not consider them in the transit time computation.

Multiple segment output
The log record for the first (or only) segment processed creates output PUOW group. The log procedure skips log records for subsequent segments.

Multiple outputs
Each output creates an output PUOW group. Multiple outputs are valid only when a corresponding input group is present.

Multiple transactions per schedule of a PSB
The log procedure normally creates the PSB group when a type X'08' log record is processed. The type X'31' input log record, which contains a PUOW and the RTKN of the PSB group, creates the connection between an input PUOW group and the corresponding PSB group. The composite record for a completed transaction is written as soon as all the type X'33' records for the OUOW group are found, but the PSB group is kept until composite records have been written for all the related transactions.

Program-to-program switch
Distinctions are made between transactions that started with a type X'01' log record (root transaction), and those that started with a type X'03' log record that has the MSGQDES flag set to X'81' (destination is an SMB, indicating a program switch to a child transaction). The log procedure retains all the OUOW group information for the root transaction until all the information for all the child transactions is complete, then writes all the transaction information to a single composite record.

AOI user-exit initiated transactions
The log procedure treats the type X'03' log record corresponding to the AOI user exit like a type X'01' log record and creates input OUOW and PUOW groups.

Output message reenqueue
When an output message is reenqueued (for example, when IMS finds that the terminal does not acknowledge successful receipt of a message) IMS will do one of the following:

1. Save the output message (indicated by a type X'36' log record)
2. Reenqueue the same message to the same destination (indicated by a second type X'35' log record with the reenqueue flag set)
3. Get the unique message from the output queue again (indicated by a second type X'31' log record)

4. Dequeue the message, if the terminal acknowledges successful receipt of the message (indicated by a second type X'36' log record)
5. Delete the message from the queue (indicated by a type X'33' log record)

The log procedure detects the output message reenqueue condition and captures all the records for this message. The record procedure extracts the date and time of the first enqueue and the date and time of the first get unique, allowing the delay to be attributed to the network.

Message-driven BMP programs
These are treated like full-function transactions.

Non-message-driven BMP programs
There are no message queue log records for these programs so they will only be used to populate the IMS_PSB_ACCOUNT_*x* tables.

System-generated output (including master terminal operator (MTO) traffic)
The output OUOW and PUOW groups are created when the type X'03' log record is processed.

Terminal message switch
The input OUOW and PUOW groups are created when the type X'01' log record is processed. This special case may also include MSC and ISC message switching.

Conversational transactions
The type X'11', X'12', and X'13' log records are not supported.

Fast Path (EMHs)
The EMH OUOW and PUOW groups are created when the type X'5901' log record is processed. The log procedure matches subsequent type X'5903', X'5936', and X'5937' log records to the EMH entries using the unit of work as a key. MSGQ output produced by EMH transactions (if any) is linked through the 16-byte recovery token in the type X'35' Out log record of the output PUOW. Full-function transactions accessing fast path databases produce type X'5937' log records. These are linked to the OUOW and PUOW groups for the full function transaction using the recovery token.

Wait-for-input (WFI) programs
If a region is processing wait-for-input (WFI) or pseudo wait-for-input (PWFI) transactions with a high PROCLIM value the type X'07' program termination log record may not be contained in the current log. In this situation you could consider reducing the PROCLIM value, or scheduling a new region so that the type X'07' log records are written more frequently. If the regions and transactions are properly classed then a 'Quick Reschedule' will occur such that the type X'07' and X'08' log records are created but the program is never truly terminated and rescheduled. The only alternative is to include all logs up to the termination of the relevant regions.

Quick reschedule
Quick reschedule is always enabled for transactions with a PROCLIM value greater than zero. It allows application programs to process more than the PROCLIM number of messages per schedule. Quick reschedule eliminates the processing overhead of rescheduling and reloading application programs. IMS uses the process limit count to ensure that no transaction type can monopolize a message region if other transactions are waiting and are eligible for processing in that region. The process limit count (PROCLIM or PLC) of a transaction specifies how many waiting

messages can be processed after the program has been scheduled and before IMS assesses whether it should be allowed to continue (quick reschedule). PROCLIM has relevance only when transactions arrive faster than they are processed (or with PWFI or WFI), so that a queue of waiting messages builds up. If PROCLIM=0, one and only one message is processed per program scheduling (no quick reschedule and no PWFI). If PROCLIM=65565, the number of messages that can be processed per scheduling is unlimited. When a quick reschedule occurs IMS may not write the type X'08' log record. In this case the log procedure creates the PSB entry when the type X'07' log record is processed. The type X'08' log record, if it follows, is matched to the already-created PSB entry.

ISC, MSC, and front-end switching (FES)

The IMS records written in this case are the same as those written for the terminal message switch case. Therefore, the log procedure treats this case exactly like a terminal message switch. For more information, see the discussion of terminal message switch on page “Terminal message switch” on page 16.

CPI-C Saa Driven Application Programs

The CPI-C program termination log record type X'0A07' is processed by the log collector to populate the IMS_PSB_ACCOUNT_*x* tables reporting resource consumption. The rows in the IMS_PSB_ACCOUNT_*x* tables resulting from The CPI-C Saa driven application program rows in these tables have the value ‘CPI REGION’ in the PROGRAM_TYPE column. A CPI-driven application program can send messages to other terminals (either LU 6.2 or non-LU 6.2) or other IMS transactions (either local or remote) by inserting an alternative PCB, after allocating the appropriate PSB. CPI-C application programs that run transactions in an MPP region are reported in the IMS_TRAN_*x* tables with the value ‘M——S–C–’ in the TRANS_TYPE key column (refer to TRANS_TYPE explanation in the “TRANS_TYPE key column” on page 94).

Synchronous APPC(OTMA) Conversations

In this case there are no type X'35' Out log records because the output does not get queued for a synchronous conversation. Instead, at application syncpoint time, IMS APPC code is called under the dependent region and it performs a GU and SEND for the output message. The log record flow is X'03'-X'31'-X'33'. The type X'31' log record is a special bypass enqueue GU record, as designated by the QLGU1NOE bit.

Release dependency

There is a different version of the log procedure and the record procedure for each version of IMS supported by the IMS CSQ feature. See the IMSVER parameter in “Log procedure parameter descriptions” on page 35 for more information.

The log procedure interprets log record layouts to determine logic flow. Because these layouts can change from one IMS release to another, the log procedure uses DSECTs from the relevant IMS release. Thus, the log procedure is largely release-independent, because the release dependency is not in the code but in the data definition. The main procedure module invokes the relevant version of a module for the IMS release specified at runtime. However, if there are changes in an IMS release that alter the log record DSECTs, an update or new release of the IMS CSQ feature with new code versions of the modules will be needed to run with the new release of IMS.

Log procedure DRLOUT reports

During normal processing, the log procedure produces informational, warning, and error messages, and some reports. For information about messages and codes issued by the log procedure, see the topic "IMS Feature Messages" in the *Messages and Problem Determination* manual.

The log procedure parameter report (Figure 3) shows the parameters in effect for this log collector run, indicating the parameters specified from the input parameter file DRLIPARM and those that used the default value.

```
DRL2071I Parameters used in this run:
------------------------------------------------------------
 Default       FLUSH TYPE                 = NONE
 Default       FLUSH MESSAGES             = NO
 Default       TABLE FLUSH                = NONE
 Default       MAX FREE POINTERS          = 800
 Default       START                      = 0000000F  0000000F
 Default       STOP                       = 2050365F  2359599F
 Default       RECTYPE                    = FF
 DRLIPARM      WRITEPENDING               = YES
 Default       SQ NLOGS                   = 1
 Default       OTMA TRAN CODE             = NO
 Default       SECONDARY TRANSACTIONS     = YES
 Default       IMS 07 RECORD BUFFER SIZE  = 200000
 Default       STATISTIC                  = YES
 Default       ACCOUNT                    = YES
 ------------------------------------------------------------
```

Figure 3. Example of IMS CSQ feature log procedure parameter report

The log procedure node statistics report (Figure 4 on page 19) appears after the log procedure has completed and provides information about the allocation and usage of "nodes" in internal tables by type. Nodes are the internal representation of data before it is grouped into composite records.

When the collect process encounters an `X'4001'` record indicating an IMS COLD start, any incomplete table entries need to be processed before continuing with new IMS records because there will be no more information about existing inflight transactions. If `WRITEPENDING=YES` was specified in DRLIPARM, the pending nodes are written as incomplete composite records. If `WRITEPENDING=NO` is specified or defaulted, the pending data is discarded.

When the collect process finishes, some nodes may be left pending because the information required to complete the composite record is not present in the log. The log procedure writes the pending nodes to the checkpoint file allocated to DRLICHKO (if present), which can be used when processing the next log for the same IMS system. If DRLICHKO is not allocated and WRITEPENDING=YES was specified in DRLIPARM, then the pending nodes are written as incomplete composite records. If DRLICHKO is not allocated and WRITEPENDING=NO is specified or defaulted, then the pending data is discarded.

The node statistics report also includes the highest "max free pointer used". If this value is close to the specified or defaulted DRLIPARM MAXFREE value for the collect, then you should consider using a higher value. The collect step terminates if the MAXFREE value is reached and more internal table storage needs to be allocated.

```
+--------------------------------------------------------------+
DRL2072I Statistics for NODEs created this run:
 --------------------------------------------------------------
 NODE          NODE  Initially     Total      Total      NODEs
 Type        length  allocated  allocated  allocated    pending
                      (NODEs)    (NODEs)    (bytes)
 --------------------------------------------------------------
 OUOW            80       4000       4000     320000         41
 PUOW            80      16000      16000    1280000        116
 PSB             48       4000       4000     192000         18
 IMS             80      80000      80000    6400000        351
 SET07BUF       124      20000      20000    2480000          0
 --------------------------------------------------------------
 TOTALS                 124000     124000   10672000        526
 --------------------------------------------------------------
 Highest Max Free Pointer used =        5
+--------------------------------------------------------------+
```

Figure 4. Example of IMS CSQ feature log procedure node statistics report

The log procedure SET07BUF buffer usage statistics report Figure 5 follows the node statistics report and provides information about the buffer used to hold IMS type X'07' program termination log records. This report can be used in conjunction with the SET07BUF figures in the node statistics report to help choose an appropriate value for the SET07BUF DRLIPARM parameter.

```
DRL2087I Buffer usage statistics for IMS 07 record buffer Set07Buf.
         Maximum number of Set07Buf entries searched = 1
         Minimum number of Set07Buf entries searched = 1
         Average number of Set07Buf entries searched = 1
         Number of times Set07Buf Searched          = 68
         Number of times 07 record not found        = 0
         Number of Set07Buf entries used            = 94832
         Number of Set07Buf hash table slots        = 32707
         Number of records written to DRLTMP07      = 0
```

Figure 5. Example of buffer usage statistics for the IMS 07 record buffer

Note: The DRLTMP07 record count only appears if the DRLTMP07 DDname is allocated in the collect JCL.

- **If you are not using a DRLTMP07 data set to buffer IMS type X'07' log records:**

  The most important statistic in the report is the average number of Set07Buf entries searched. If this value increases and an associated increase in collect CPU time for the same size log is observed, then you should consider increasing the value for SET07BUF, using the total allocated nodes figure from the node statistics report as a guide. To be effective in improving performance an increase in the SET07BUF parameter should be associated with an increase in the reported number of Set07Buf hash table slots.

  If the number of times the 07 record is not found is large then the transaction data stored by Tivoli Decision Support for z/OS may be missing program-specific information for an unacceptable number of transactions. If you are processing your logs in small quantities then you may be able to reduce this number by processing more logs in each collect process.

- **If you are using a DRLTMP07 data set to buffer IMS type X'07' log records:**

  The same considerations apply to the average number of Set07Buf entries searched figure as for when DRLTMP07 is not used.

  If the number of times the 07 record is not found is large then the transaction data stored by Tivoli Decision Support for z/OS may be missing program-specific information for an unacceptable number of transactions. In this case you should consider increasing the SET07BUF setting to allow more type X'07' log record data to be held in memory. The number of records written to DRLTMP07 can be used to help size the allocation of the DRLTMP07 data set.

Note that the entire IMS type X'07' log record is written to DRLTMP07, so you must use the appropriate record length for the IMS system producing the log when allocating DRLTMP07.

Refer to the description of SET07BUF in Chapter 4, “Administering the IMS CSQ feature,” on page 29 for more information.

The log procedure table flush processing statistics report (Figure 6) follows the SET07BUF statistics report if table flush processing was requested. It provides information about how often table flush processing was invoked, and about the data that was flushed from the log procedure internal tables.

```
DRL2084I TABLEFLUSH processing statistics:
  Number of times TABLEFLUSH invoked           = 305
  Number of times TABLEFLUSH invoked for SOS    = 0
  Total number of OUOWs flushed                = 389
  Total number of PUOWs flushed                = 230243
  Total number of input PUOWs flushed          = 230078
  Total number of completed input PUOWs flushed = 229812
  Average storage freed per table flush        = 0.3 Mb
```

Figure 6. Example of TABLEFLUSH processing statistics

If table flush processing was invoked one or more times for a short-on-storage condition (SOS) you should consider increasing the amount of virtual storage available to the collect process, increasing the frequency of table flush processing, or reducing the "age" at which table entries are flushed.

You can use the ratio of "completed input PUOWs flushed" to "input PUOWs flushed" as an indication of how much data is being affected by the flush process. Table flush processing can produce duplicate transactions and incomplete transaction records, but if the proportion of completed input PUOWs is high the impact on the aggregated data stored in the product database may be small enough to be justified by the associated improvement in collect performance.

When considering the impact of table flush processing, be aware that the table flush statistics only relate to records written during the flush process. This means that the impact is generally much less than implied by the ratio between completed and total input PUOWs flushed.

Note: Aggressive table flush settings may result in null values in key columns when the log procedures are used to collect data into the TDS/z database. Message DRL0336W is issued for each affected table.

Refer to the description of FLUSHTYPE and TABLEFLUSH in Chapter 4, “Administering the IMS CSQ feature,” on page 29 for more information.

Log procedure write pending processing messages Figure 7 follow the SET07BUF statistics or the table flush statistics (if present). These messages provide information about the data that was written from the log procedure internal tables.

```
DRL2085I WRITEPENDING processing started after 3054628 records.
DRL2086I WRITEPENDING complete: 19 OUOWs written as composite records.
        6985 input PUOWs flushed (6981 completed).
```

Figure 7. Example of WRITEPENDING processing messages

You can use the ratio of "completed" input PUOWs flushed to "input PUOWs flushed" as an indication of how much of the pending data is incomplete. If the proportion of completed input PUOWs is high the impact on the aggregated data stored in the Tivoli Decision Support for z/OS database may be small enough to be justify including all the log data by using the write pending process.

Alternatively you can allocate a DRLICHKO data set and preserve the pending data for processing in a subsequent collect step.

Refer to the description of WRITEPENDING in Chapter 4, “Administering the IMS CSQ feature,” on page 29 for more information about the write pending process. Refer to the description of the DRLICHKO and DRLICHKI data sets in Chapter 4, “Administering the IMS CSQ feature,” on page 29 for more information about using the checkpoint facility.

The log procedure composite record statistics report Figure 8 follows the SET07BUF statistics report, the table flush statistics report, or the write pending processing messages, depending on the combination of DRLIPARM parameters specified. It provides information about the composite records written by the log procedure.

```
DRL20731 Statistics for composite records created this run:
---------------------------------------------------------------------------------
                                                          Min    Max       Total
Record                                           Count    Lth    Lth       Bytes
---------------------------------------------------------------------------------
 01  Transaction data                             5267    404   26564   123507100
 07  Program termination                         94832    367     367    34803344
 08  Program schedule                            94832    131     131    12422992
 40  Checkpoint                                      3   1509    1509        4527
 45  Checkpoint statistics                         104     47    3771       55084
 47  Checkpoint region active                       49     83    1213       54553
 ---------------------------------------------------------------------------------
                                                195087     47   26564   170847600
----------------------------------------------------------------------------------
```

Figure 8. Example of composite record statistics

The record procedure

The IMS CSQ feature log procedure produces composite records containing transaction information extracted from the raw IMS log records. Each composite record contains information for a primary transaction and any secondary transactions it initiated. The composite records are processed by a record procedure to produce one record for each transaction. These transaction records are called R2 records, and are processed by Tivoli Decision Support record and update definitions to update the IMS CSQ feature transaction tables.

The record procedure

Chapter 3. Understanding data flow through the IMS CSQ feature

This topic describes the flow of data through the IMS CSQ feature. The topic includes:

- "Log collector data flow"
- "DRLSLOGP data flow" on page 24
- "IMS CSQ light feature data flow" on page 26
- "IMS CSQ feature object definitions" on page 26

Log collector data flow

Figure 9 shows the flow of data from the SLDS to the DB2 tables when you use the IMS CSQ log collector. For more information about running the log collector, see "Running the log collector" on page 31.

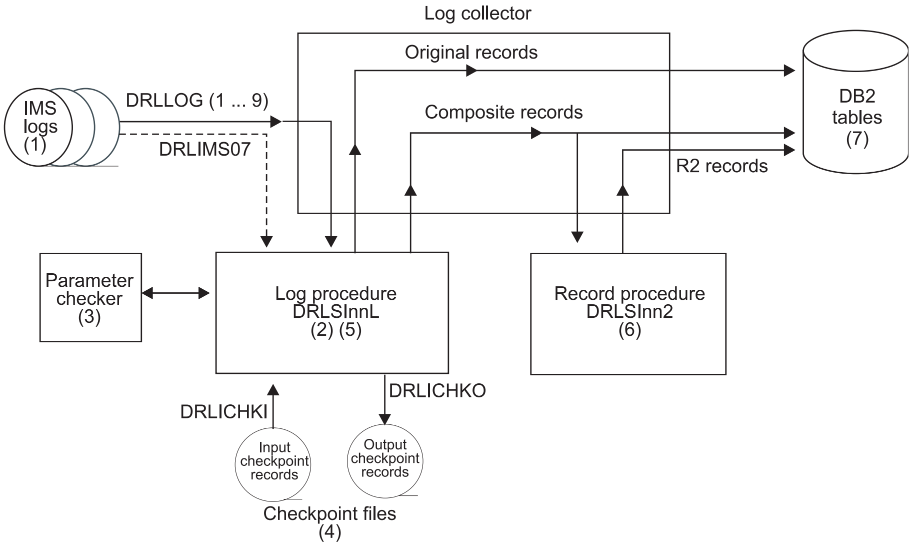


Figure 9. IMS CSQ data flow through the log collector

The log collector performs the following steps to update the database tables:

1. The IMS logs contain the original data as written by IMS, or the log that was created by merging the SLDSs written by IMS systems that are members in an IMS shared group. The DRLLOG (1 ... 9) DDNAMEs point to the data sets you want to process. You can optionally preprocess the IMS log data sets and extract all the type X'07' log records. These records can then be input to the log procedure via the DRLIMS07 DDNAME.
2. The log collector calls the log procedure (DRLSI*nn*L, where *nn* is the level of IMS you are using; for example, DRLSID1L for IMS version 13, release 1).
3. The log procedure calls a parameter-checking procedure to verify the processing options you selected. If you do not specify any parameters, the log procedure uses the default parameters. The DDNAME is DRLIPARM, which points to the parameter data set or contains in-stream parameters.
4. The DRLICHKI and DRLICHKO DDNAMEs are used to hold pending IMS activity. At key commit times while the log collector is running, the log

procedure writes to the checkpoint file referred to by the DRLICHKO DDNAME. You can use this file to restart in the event of a failure or when processing the next log. After the log procedure finishes processing, it writes the remaining unmatched transaction records to DRLICHKO. Allocate the file output to DRLICHKO to DRLICHKI the next time you use the log procedure.

At the start of processing, the log procedure checks for the file referred to by the DRLICHKI DDNAME. If DRLICHKI is present, it contains unmatched transaction records from the last time the log collector was run. The log procedure can complete the composite records for these transactions with the new IMS input log.

5. If the DRLIMS07 file is not used, the log procedure performs a "first pass" over the log data to extract and buffer the type X'07' log records. If DRLIMS07 is used, the type X'07' log records are read directly from the file to the buffer. If the DRLTMP07 file is used, the buffer will be loaded with X'07' records up to the limit set by the SET07BUF parameter, and any additional X'07' records will be written to the DRLTMP07 file and moved from the file to the buffer at a later stage (when storage is freed). If the DRLTMP07 file is not allocated, then all the X'07' records are placed in the buffer irrespective of the SET07BUF setting. In situations where a log contains too many X'07' records to be held in memory, the COLLECT may fail with a storage related error. When the buffer has been loaded, the log procedure performs a "second pass" over the log data to extract the information stored by Tivoli Decision Support. The log procedure does all of the matching and processing to create the composite records. It sends the composite records and, optionally, the original records back to the log collector.
6. The log collector sends the composite records to the record procedure (DRLSI*nn*2, where *nn* is the level of IMS you are using; for example, DRLSID12 for IMS version 13, release 1) where the record procedure simplifies the records for easier collection and reporting.
7. The log collector uses the R2 records from the record procedure and the composite records from the log procedure to update DB2 tables.

DRLSLOGP data flow

Figure 10 on page 25 shows the flow of data from the SLDS to the output records when you use DRLSLOGP. For information about running DRLSLOGP, see “Using DRLSLOGP” on page 39.

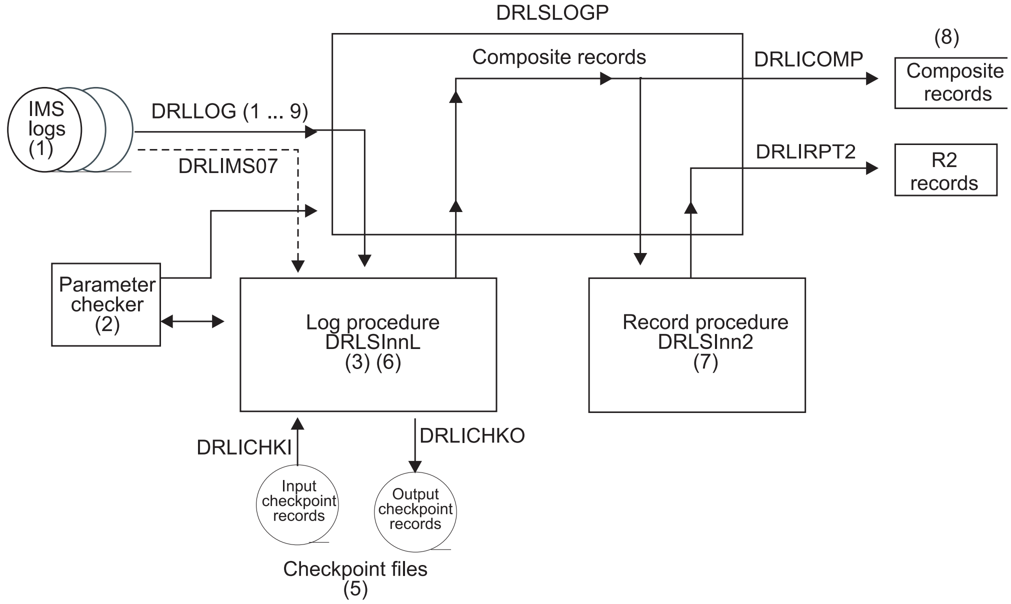


Figure 10. Data flow through DRLSLOGP

DRLSLOGP creates composite records or R2 records by following this process:

1. The IMS logs contain the original data as written by IMS, or the log that was created by merging the SLDSs written by IMS systems that are members in an IMS shared group. The DRLLOG (1 ... 9) DDNAMEs point to the data sets you want to process. You can optionally preprocess the IMS log data sets and extract all the type X'07' log records. These records can then be input to the log procedure via the DRLIMS07 DDNAME.
2. DRLSLOGP calls a parameter-checking procedure to check the processing options you have selected. If you do not specify any parameters, DRLSLOGP uses the defaults. The DDNAME is DRLIPARM, which points to the parameter data set or contains in-stream parameters.
3. DRLSLOGP calls the log procedure (DRLSI*nn*L, where *nn* is the level of IMS you are using; for example, DRLSID1L for IMS version 13, release 1).
4. The log procedure calls a parameter-checking procedure to verify the processing options you selected. If you do not specify any parameters, the log procedure uses the default parameters. The DDNAME is DRLIPARM, which points to the parameter data set or contains in-stream parameters.
5. The DRLICHKI and DRLICHKO DDNAMEs are used to hold pending IMS activity. After the log procedure finishes processing, it writes the remaining unmatched transaction records to DRLICHKO. Allocate DRLICHKO to DRLICHKI the next time you use the log procedure.

   The log procedure also checks for the existence of checkpoint file DRLICHKI. If DRLICHKI is present, it contains unmatched transaction records from the last time the log collector was run. The log procedure can complete the composite records for these transactions with the new IMS input log.
6. If the DRLIMS07 file is not used, the log procedure performs a "first pass" over the log data to extract and buffer the type X'07' log records. If DRLIMS07 is used, the type X'07' log records are read directly from the file to the buffer. If the DRLTMP07 file is used, the buffer will be loaded with X'07' records up to the limit set by the SET07BUF parameter, and any additional X'07' records will be written to the DRLTMP07 file and moved from the file to the buffer at a later stage (when storage is freed). If the DRLTMP07 file is not allocated, then

all the X'07' records are placed in the buffer irrespective of the SET07BUF setting. When the buffer has been loaded, the log procedure performs a "second pass" over the log data to extract the information stored by Tivoli Decision Support. The log procedure does all of the matching and processing to create the composite records, and sends the composite records back to DRLSLOGP.

7. DRLSLOGP sends the composite records to the record procedure (DRLSI*nn*2, where *nn* is the level of IMS you are using; for example, DRLSID12 for IMS version 13, release 1) which can simplify the records for easier collection and reporting. The record procedure writes the output to DDNAME DRLIRPT2. There could be one or more R2 records produced by the record procedure for the same composite record. You can specify this output file as disk, tape, or dummy output.
8. DRLSLOGP sends the composite records to the DRLICOMP file, which you can specify as disk, tape, or dummy output.
9. Depending on the REPORTS options requested, the data sets DRLIRPT2 and DRLICOMP may also contain Transaction Level Statistics (56FA) records. Additionally, there may be the optional data set DRLIRPT0, which contains only Transaction Level Statistics (56FA) records.

IMS CSQ light feature data flow

If you use the IMS light feature, the initial data flow at the remote site is the same as for DRLSLOGP (see "DRLSLOGP data flow" on page 24), from the IMS logs to DRLIRPT2. When you collect the R2 records at the central Tivoli Decision Support database site, the data flow is as shown in Figure 11.

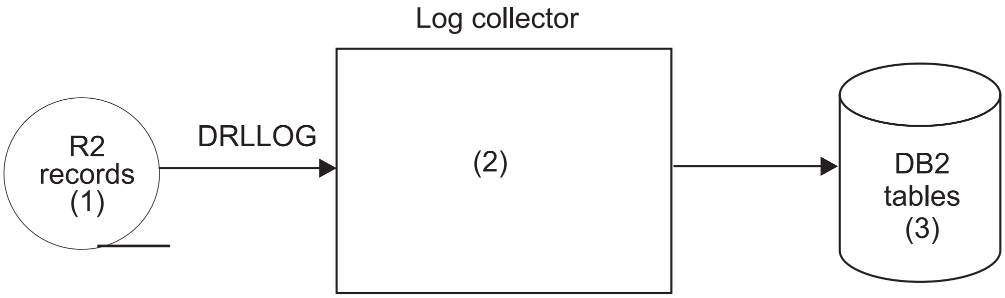


Figure 11. Data flow through IMS CSQ light feature

The light log collector does the following to update the database tables:

1. The R2 input contains the R2 extract records created at the remote site by running DRLSLOGP. The DRLLOG DDNAME points to the data set you want to process.
2. The log collector uses the IMS CSQ Light log definition to process the R2 records.
3. DB2 tables are updated with data from the R2 records.

IMS CSQ feature object definitions

Figure 12 on page 27 and Figure 13 on page 28 show the flow of data through the IMS CSQ feature objects when you use the log collector. They show the data flow from the original IMS log data to the IMS CSQ feature reports.

The characters *nn* in the figure indicate the IMS release, and can be B (IMS version 11), C (IMS version 12), or D (IMS version 13); the characters `nnn` also indicate the

IMS release, and can be B (IMS version 11), C (IMS version 12), or D (IMS version 13).

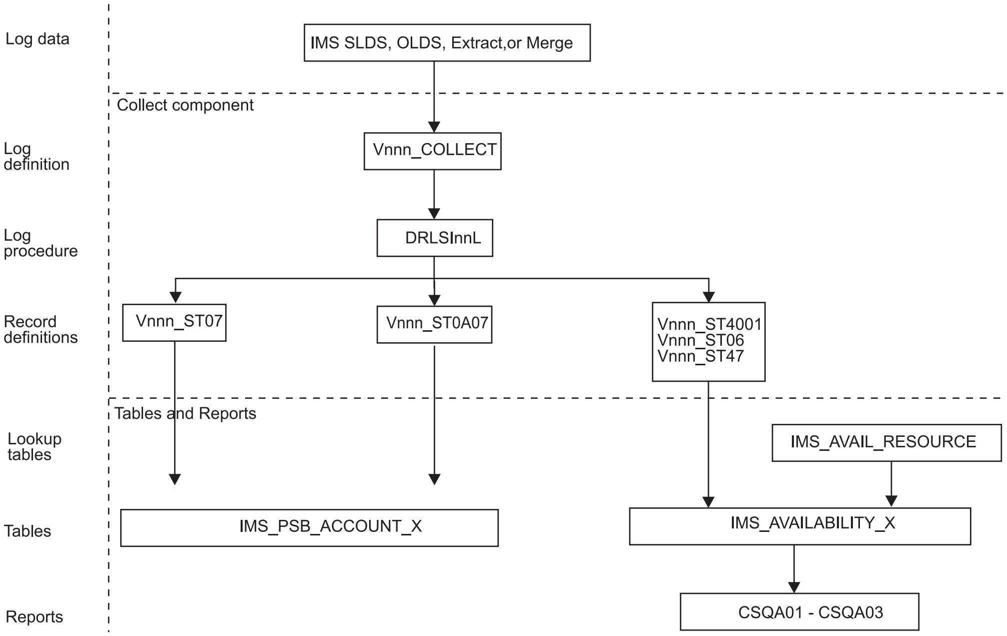


Figure 12. Data flow: Account and Availability subcomponent

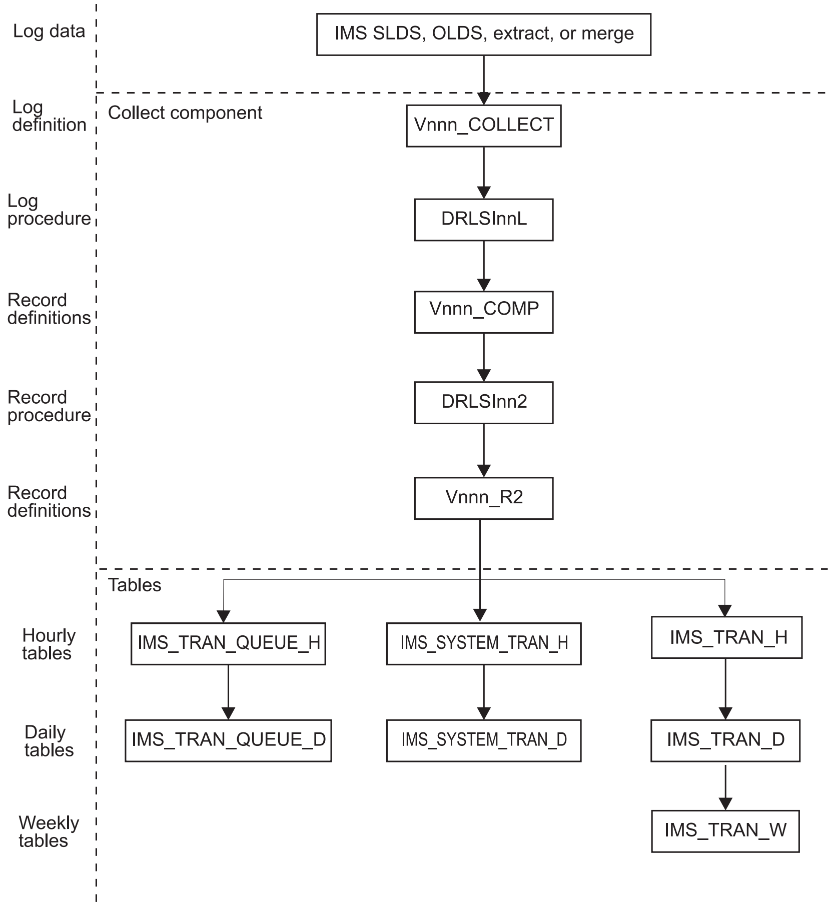


Figure 13. Data flow: Transaction Transit Time subcomponent

Chapter 4. Administering the IMS CSQ feature

This topic explains how to use the IMS CSQ performance feature to process and collect IMS data. You can use the log collector program when collecting data into DB2 tables or you can use DRLSLOGP in batch mode when generating composite records. For more information about these, see Chapter 3, “Understanding data flow through the IMS CSQ feature,” on page 23.

IMS logs in a shared-queue environment

In a SYSPLEX IMS environment, each IMS system has its own system log data set (SLDS) where transaction records are logged.Tivoli Decision Support for z/OS reads the log records from the SLDS file, recreates the transaction flow, and evaluates the IMS statistics, saving the data into the DB2 tables of the IMS CSQ feature.

When working in a shared-queue environment, each IMS continues to write its own log so that the records for a 'unique' transaction can be spread across several SLDS files. Tivoli Decision Support for z/OS needs to use all the SLDS logs to re-create the complete transaction across the IMS shared-queue environment.

Within an IMS shared-queue environment, Tivoli Decision Support for z/OS provides the following options:

1. Merge all logs using DFSLTMG0, the merge utility provided with IMS. This processes the IMS logs before the collect is run. The maximum number of logs merged by the IMS utility is nine. For more information about this, see Appendix D, “DFSLTMG0 log merge utility,” on page 157.
2. Utilize the product's internal merge option by specifying the SQNLOGS parameter. This opens all the logs from the IMS systems defined in the collect JCL and then reads, in turn, the first record of each log and decides which one is the next record to be processed. The maximum number of logs that can be merged internally like this is nine. For more information, see “Running the log collector” on page 31.

In all cases TDS can only process the logs together if all the IMS systems are the same version. The log data should all refer to the same time period.

Using the log collector

The log collector runs the IMS CSQ feature log procedure in a Tivoli Decision Support for z/OS environment. It uses a log procedure and record procedure, invokes DB2 and updates DB2 tables.

Input and output data sets

The log collector has the following inputs and outputs (listed here by DDNAME):

DRLLOG - input IMS logs

The input IMS log data. The input log is usually the SLDS or an appropriate extract, but you can use the OLDS after IMS has closed it. If your IMS systems are running in a shared-queue environment, merge all the logs produced by each IMS system that is a member of a shared-queue environment, to build a unique log to use for the collect. The input log can be a data set built by merging the IMS SLDSs from all the IMS systems that are members of the same IMS shared group. This can be obtained by using the merge utility DFSLTMG0 provided by IMS or by using the online merging-option (SQNLOGS parameter). You can also process each log separately but some data, such as timing information, might become inaccurate. If the online merge option is used, specify DRLLOG1...n.

DRLOUT - output messages
The IMS CSQ performancefeature writes messages to this DDNAME. You can allocate this DDNAME to SYSOUT, a physical data set, or dummy.

DRLDUMP - output error information
The IMS CSQ feature writes error information to this DDNAME. You can allocate this DDNAME to SYSOUT, a physical data set, or dummy.

DRLICHKI - input log procedure checkpoint file (optional)
The input data set that contains the status of all pending IMS activities written when the log procedure completed processing on a previous run. This data set ensures that the IMS CSQ support can process IMS log data in discrete data set level parts without loss of data. The layout is internal. You can allocate this DDNAME to DUMMY or to a previously created output log procedure checkpoint file. Alternatively, you can omit it.

DRLICHKO - output log procedure checkpoint file (optional)
The output data set that records the status of all pending IMS activities when the log procedure completes processing the current log data. This data set can be processed later by the log procedure, if it is allocated to the DRLICHKI DDNAME. This data set can be large, for a large IMS system with many secondary transactions. You can allocate this DDNAME to DUMMY, allocate it to a data set with record length of 32 756 bytes, or omit it.

The following DDNAMEs are used by the IMS CSQ feature log procedure:

DRLIPARM - input log procedure parameters
This DDNAME points to the parameter data set or contains the in-stream parameters. See “Specifying log procedure parameters” on page 33 for more information.

DRLIMS07 - extracted X'07' records
IMS shared-queue processing buffers type X'07' records to make them available for processing when required. IMS type X'07' records are normally read from the IMS logs in the first pass of a two-pass process. However, an external facility, such as IBM DFSORT, can be used to replace the first pass by extracting the IMS X'07' records before running the collect job. This can improve performance by reducing the amount of time to read the logs.

If the IMS X'07' records have been extracted in this way, specify the data set holding these records in the DRLIMS07 DDNAME of the collect job; the collect then takes the IMS type X'07' records from this data set.

If you use the DRLIMS07 DDNAME for input you must ensure that it contains only type X'07' records, and that the records are in the same sequence as they appear in the related IMS logs. If you are processing multiple IMS logs from different systems, then merge the logs using DFSLTMG0 before extracting the type X'07' records.

DRLTMP07 - temporary X'07' records
IMS shared-queue processing buffers the type X'07' records to make them

available for processing when required. IMS type X'07' records are normally read from the IMS logs in the first pass of a two-pass process. In situations where a log contains too many X'07' records to be held in memory, the COLLECT may fail with a storage related error. To resolve this, the DRLTMP07 file can be used to contain the 'excess' records. When the DRLTMP07 file is defined as a DDNAME in the JCL, the SET07BUF parameter will be used to identify the maximum number of records to be stored in memory. Additional X'07' records over and above the SET07BUF parameter setting will be written to the DRLTMP07 dataset. During the second stage once an X'07' record is freed from the SET07BUF it will be replaced by the X'07' from the DRLTMP07 dataset.

To use this DDNAME, allocate it in the JCL with the following attributes:

```
//* If using Tape include DCB BLKSIZE parameter with a size suitable
//* for your installation.
//DRLTMP07 DD DCB=(LRECL=456,RECFM=VB),
//         DSN=&DRLTMP07,
//         SPACE=(456,(1000,200),RLSE),AVGREC=K,
//         VOL=(,,,3),
//         DISP=(NEW,DELETE)
```

This sample allocates DRLTMP07 as a temporary data set that is deleted at the end of the collect step, with up to three volumes requested for allocation. The logical record length specified (LRECL) must be at least four more than the length of the type X'07' records in your IMS log. This value is dependent on the IMS version and maintenance level. You can set the minimum required value to minimize virtual storage use. Alternatively you can specify a large value like 32756 to cater for all versions of IMS and allow for future changes, at a small cost in virtual storage use.

The SPACE parameter reflects the number of type X'07' records you expect to process in a typical log. Note that the logical record length here is only an estimate and does not have to match the DCB LRECL parameter or exactly match the true record length. The sample allocation anticipates 1 million IMS V10 type X'07' log records with considerable margin for error in the secondary space parameter. Given that a typical log comprises between two and five per cent type X'07' records, this allocation is sufficient for a log containing in the order of 20 million records.

Running the log collector

About this task

You can run an online collect from the administration dialog, or in batch. To run a collect, refer to the *Administration Guide and Reference*. .

Figure 14 on page 32 shows an example of how to run an IMS CSQ feature collect in batch.

```
//USERIDA   JOB (ACC000,001),'IMS1 COLLECT',
//          NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//COLLECT   EXEC PGM=DRLPLC,
//   PARM=('SYSTEM=DB21',
//         'SYSPREFIX=DRLSYS',
//         '&PREFIX=DRL',
//         'SHOWSQL=NO',
//         'SHOWINPUT=NO')
//STEPLIB   DD DISP=SHR,DSN= Tivoli Decision Support for z/OS load library
//          DD DISP=SHR,DSN=DB2 load library
//DRLLOG    DD DISP=SHR,DSN=(IMS SLDS or IMSes shared merged log or DRLICOMP)
//*
//DRLICHKI  DD DUMMY    -- or previously created checkpoint data set
//DRLICHKO  DD DUMMY    -- or LRECL=32756 output checkpoint data set
//*
//DRLOUT    DD SYSOUT=*,DCB=(LRECL=80)
//DRLDUMP   DD SYSOUT=*,DCB=(LRECL=32756)
//DRLIPARM  DD *
ACCOUNT=YES
MAXFREE=3000
/*
//DRLIN     DD DISP=SHR,DSN=USERIDA.IMS.DEFS.V61(DRL$CVAR)
//          DD *
COLLECT     CSQ_VB10_COLLECT      -- for IMS V11
                                  -- or CSQ_VC10_COLLECT for IMS V12
                                  -- or CSQ_VD10_COLLECT for IMS V13
            BUFFER SIZE 50 M;     -- Appropriate collect buffer size
/*
//
```

Figure 14. Sample job for running the log collector within IMS CSQ

To run the log collector for logs from different versions of IMS, change the log name specified on the collect statement.

The DRLLOG DD statement specifies the input IMS log data. If you are using logs from different IMS systems and you need to open and merge them online (in the same shared-queue group), then use DRLLOGx by specifying one DDNAME statement for each different IMS system working in the shared-queue environment.

For example, if you have four IMS working in a shared-queue environment, and you use the internal log merge, you must specify four different DRLLOGx DD statements, one for each IMS, to describe its logs:

```
DRLLOG1 DD DISP=SHR,DSN=...
DRLLOG2 DD DISP=SHR,DSN=...
DRLLOG3 DD DISP=SHR,DSN=...
DRLLOG4 DD DISP=SHR,DSN=...
```

and set SQNLOGS=4 in the DRLIPARM.

Note: If you are using only one log, you must specify DRLLOG DD and not DRLLOG1 DD.

Specifying the log collector parameters

About this task

When running the log collector, you can specify log collector parameters and `COLLECT` statements by using in-stream JCL or a data set allocated to DDNAME DRLIN. The format for specifying these parameters is as follows:

```
   SET parameter name='value';
```

For example, the IMS_SYSTEM_ID of IMS1 is specified as follows:

```
SET IMS_SYSTEM_ID = 'IMS1';
```

Refer to the *Language Guide and Reference* for a description of the COLLECT and SET statements.

The following parameters are applicable to the IMS collect:

```
IMS_SYSTEM_ID   - IMS system ID
SYSPLEX_NAME    - sysplex name
MVS_SYSTEM_ID   - MVS system ID
IMS_CTRL_REGION - IMS control region
IMS_APPLID      - VTAM application ID name
```

You can use TDSz to specify only a single value for each of these parameters, and therefore it is not advisable to use these parameters when you are collecting data from multiple IMS systems in a shared queue environment. In this case you will need to populate lookup table IMS_SYSTEM_NAMES with your IMS system IDs.

The following steps describe how TDSz populates column IMS_SYSTEM_ID during the collect:

- TDSz derives IMS_SYSTEM_ID from the IMS records.
- When IMS_SYSTEM_ID cannot be derived from the IMS records, column IMS_SYSTEM_ID is populated based on the following conditions:
  – When IMS_SYSTEM_ID is specified with the SET statement, IMS_SYSTEM_ID is used to populate column IMS_SYSTEM_ID.
  – When IMS_SYSTEM_ID is not specified with the SET statement, column IMS_SYSTEM_ID is set to '$UNKNOWN'.

The following steps describe how TDSz populates columns SYSPLEX_NAME, MVS_SYSTEM_ID, IMS_CTRL_REGION, and IMS_APPLID during the collect:

- TDSz uses lookup table IMS_SYSTEM_NAMES to obtain values for these columns.
- When values are not available for these columns in lookup table IMS_SYSTEM_NAMES, these columns are populated based on the following conditions:
  – When values are specified with the SET statement, these values are used to populate these columns.
  – When values are not specified with the SET statement, the columns are set to '$UNKNOWN'.

Note: To use the IMS_SYSTEM_NAMES lookup table for augmenting TDSz data tables, you must collect raw IMS log records, and not the DRLIMS07 data set to input the preprocessed X'07' records.

Specifying log procedure parameters

About this task

You can specify parameters to control the operation of the log procedure. Table 2 on page 34 lists these parameters with a brief description, the default value, and impact on collect performance. For a full description of the parameters, see "Log procedure parameter descriptions" on page 35.

You can specify the parameters in a data set or in the in-stream JCL for DDNAME DRLIPARM. All parameters must start in column 1. There must be an equal sign

(=) between the parameter and the value, with no spaces between. Quotation marks, ending colons, or semicolons are not allowed. For example, `ACCOUNT=YES`.

Table 2. Parameter summary for the IMS shared-queue log procedure

Parameter name	Possible values	Default	Description	Performance impact
ACCOUNT	YES NO	YES	Account records written to the composite record stream.	Low
FLUSHMSGS	YES NO	NO	Controls whether the log procedure issues messages each time table flush processing is invoked.	None
FLUSHTYPE	SOS, CHKPT or COUNT=*n*, where *n* is a whole number greater than or equal to one.	CHKPT if TABLEFLUSH specified, otherwise none	Controls when the log procedure invokes table flush processing.	High
MAXFREE	Whole numbers between 4 and maximum system capacity	800	Number of internal buffer pointers.	High
MSGTEXT	start - whole numbers from 1 to 255 length - whole numbers from 1 to 60	None	Collect data from the MSGXSTXT field of the IMS X'01' and X'03' records.	Low
OTMATRANCODE	YES NO	NO	Source of OTMA transaction name.	Low
PASSLOGREC	YES NO	YES	IMS log records passed as output from the log procedure.	Low
PHASE2ENDTIME	YES NO	NO	Use data from the IMS 5612 record.	Low
RECTYPE	Valid hex numbers from 00 to FF	FF	Record type of composite records.	None
SECONDARY	YES NO	YES	Process child transactions.	Low - High (depending on the number of children for each transaction)
SET07BUF	Positive integer	200000	Number of X'07' records.	None
SQNLOGS	Integer numbers between 1 and 9	1	Number of logs opened as input.	Medium - High (according to the number of logs and record sequences)
START	Valid values for *yyyy-mm-dd-hh.mm.ss.t*	Start of log	IMS log date and time to start processing.	None
STATISTIC	YES NO	YES	Records written in the DRLICOMP data set as CSQ_V*nn*_ST*xx* records.	Low
STOP	Valid values for *yyyy-mm-dd-hh.mm.ss.t*	End of log	IMS log date and time to stop processing.	None
TABLEFLUSH	Non-negative integers	1 if FLUSHTYPE specified, otherwise none	Age limit (in seconds) of table entries.	Low - High (depending on the log contents and FLUSHTYPE setting)

Table 2. Parameter summary for the IMS shared-queue log procedure (continued)

Parameter name	Possible values	Default	Description	Performance impact
VERIFY	NONE WARN FAIL	NONE	Controls log procedure data verification and determines whether the log procedure continues when an issue is identified.	None
WRITEPENDING	NO YES	NO	Pending table entries written as output.	Low

Log procedure parameter descriptions

The parameters for the log procedure within an IMS shared-queue environment are:

ACCOUNT=*xxx*

Specifies that X'06', X'07', X'08', and X'0A07' records will be written to the composite record stream as CSQ_V*nn*_ST*xx* records. If you do not install the Account and Availability subcomponent, set ACCOUNT=NO. The default is YES.

FLUSHMSGS=*xxx*

Specifies whether the log procedure issues messages each time table flush processing is invoked. If table flush processing is requested by the FLUSHTYPE or TABLEFLUSH parameters and FLUSHMSGS=YES is specified, then the log procedure issues messages at the beginning and end of each invocation of table flush processing. These messages provide information about the frequency and effectiveness of table flush processing.

In most cases the information provided in the table flush processing statistics report (message DRL2084I) is sufficient. The additional level of detail provided by specifying FLUSHMSGS=YES may be useful for IBM service personnel diagnosing log collection problems, but is not required in normal circumstances. It can significantly increase the size of DRLOUT if regular flushing is requested with a small interval using the FLUSHTYPE=COUNT=nnnnnnnn DRLIPARM parameter.

Possible values are YES and NO. The default is NO.

Note: Specifying FLUSHMSGS=NO does not suppress the table flush processing statistics report (message DRL2084I) produced when the log procedure completes, nor does it suppress the flush messages produced during write pending processing. Specifying FLUSHMSGS=YES does not cause table flush processing to be invoked unless FLUSHTYPE or TABLEFLUSH is also specified.

FLUSHTYPE=*xxxxx* or FLUSHTYPE=COUNT=*nnnnnnnn*

Specifies the circumstances in which the log procedure invokes table flush processing.

Here are the possible values. The default is CHKPT if TABLEFLUSH is specified. There is no default if TABLEFLUSH is not specified.

Table 3. FLUSHTYPE values

Value	Description
SOS	Only invoke table flush processing when a short-on-storage condition occurs. Use this value to prevent the collect process from terminating when insufficient virtual storage is available to continue processing the log.
CHKPT	Invoke table flush processing whenever an IMS checkpoint log record (type x'4001) is processed, or when a short-on-storage condition occurs. Use this value to minimise storage overhead and improve performance by invoking table flush processing on a regular basis determined by the frequency with which system checkpoints are taken in the IMS system that produced the log being processed.
Count=nnnnnnnn	Invoke table flush processing for every nnnnnnnn input log records processed, or when a short-on-storage condition occurs. Use this value to minimise storage overhead and improve performance by invoking table flush processing on a regular basis independent of the frequency with which system checkpoints are taken in the IMS system that produced the log being processed.

Refer to the description of the TABLEFLUSH parameter for more information about table flush processing.

MAXFREE=*nnnn*

Specifies a limit on the number of internal pointers that will be used to address buffers for holding incomplete transactions. If message DRL2021W is issued indicating a buffer array pointer shortage, increase the value of MAXFREE in 50% increments until the problem is resolved. Possible values are from 4 to the maximum allowed by available virtual storage. The default is 800. Specifying a small value for MAXFREE is not recommended because only buffer pointer storage is allocated based on this parameter. The transaction data buffer storage addressed by the pointers is not allocated until it is required.

MSGTEXT=*start, length*

Optional. Enables up to 60 bytes of contiguous data to be collected from the MSGXSTXT field of the IMS type X'01' and X'03' input log records. The default is to not collect data from the MSGXSTXT field.

start is the starting position of the data to be extracted from the MSGXSTXT field. The maximum value for ***start*** is 255 and the minimum is 1.

length is the length of the date to be extracted from the MSGXSTXT field. The maximum value for ***length*** is 60 and the minimum is 1.

To store the extracted data into DB2 you will need to customize your IMS feature tables. For an example see Appendix E, "Tailoring example using MSGTEXT for IMS_TRAN_x tables," on page 159.

OTMATRANCODE=*xxx*

Specifies whether the transaction name for OTMA transactions should be taken from OTMA point of view (LUY_TRANCODE of APPC SECTION) or from IMS point of view (MSGODSTN). The default value is NO, meaning that the transaction name is equal to MSGODSTN.

PASSLOGREC=*xxx*

Specifies whether the original IMS log records are passed as output from the log procedure. Possible values are YES and NO. The default is YES.

Specify NO if you:
- Do not collect logs that contain composite records.

- Do not want to use your own update definitions based on IMS log records.

PHASE2ENDTIME=*xxx*

Optional. Specifies whether data from the IMS X'5612' log records is collected into the composite and R2 records. Possible values are YES and NO. The default is NO. Specify YES if you want to collect data from the IMS X'5612' records.

This parameter causes Tivoli Decision Support for z/OS to collect data from the IMS X'5612' records (External Subsystem - End of Phase 2 Syncpoint (DBCTL)) and store it in the composite record for the unit of work. If available, the timestamp from these records is used when calculating the value for DRLMPROCE in the R2 record. This field is used to set the PROCESS_SEC column in the IMS_TRAN_H, IMS_SYSTEM_TRAN_H and IMS_SYSTEM_TRAN2_H tables.

Note: If DRLSLOGP is run with PHASE2ENDTIME=YES and the composite or R2 records output from DRLSLOGP are subsequently input to DRLPLC, specifying PHASE2ENDTIME=NO on the DRLPLC job, will not prevent the IMS X'5612' data collected by DRLSLOGP from being used in the calculation of the process_sec value.

RECTYPE=*nn*

Specifies the record type of composite records. For example, if you want the composite records to have type X'FA', specify `RECTYPE=FA`. The default is FF.

SECONDARY=*xxx*

Specifies whether or not secondary (program switch) transactions are to be processed during the collect. The default is YES. When SECONDARY is set to NO, the following applies:

- Secondary transactions are not processed by the log procedure. Therefore, they are not included in the composite record and the transaction tables are not populated by secondary transaction data.
- Any outputs from secondary transactions that cannot be distinguished from the outputs of primary transactions are processed.

The IMS_PSB_ACCOUNT tables, which are populated by X'08' and X'07' records, are not affected by any exclusion. Hence, they continue to be populated by PSB cumulative data for primary and secondary transactions, even if SECONDARY=NO is specified.

SET07BUF=*nnnnnn*

Specifies the number of type X'07' records to be buffered in the first pass over the IMS log data set. In the ideal case, this number should be large enough for the buffer to hold all type X'07' records found in the input log data set. An appropriate value is typically in the range from two to five per cent of the number of records in the input log data set. The default is 200000. When the DRLTMP07 file is allocated in the COLLECT JCL, the SET07BUF setting specifies the maximum number of X'07' records stored in memory, and any X'07' records over and above this setting are stored temporarily in the DRLTMP07 file.

If the DRLTMP07 file is not allocated in the COLLECT JCL, the SET07BUF is used as an indicator of the expected number of X'07' records and is used to improve the performance of memory allocation. In this case the SET07BUF setting does not limit the number of X'07' records in the buffer.

Thus, if a log has more X'07' records than is specified in the SET07BUF parameter, the log procedure will attempt to allocate a buffer large enough to hold all the X'07' records.

SQNLOGS=n
Specifies the number of IMS logs to be opened for input. This number should match the number of DRLLOG(n) DD statements specified in the collect or DRLSLOGP job. Valid values are between 1 and 9. The default is 1.

START=*yyyy–mm–dd–hh.mm.ss.t*
The transaction date and time starting point for processing, where *yyyy–mm–dd–hh.mm.ss.t* is the year, month, day, hour minute, second, and tenth of a second timestamp. The default is the beginning of the log.

STATISTIC=*xxx*
Specifies that X'45' records will be written into the composite record stream as CSQ_Vnn_STxx records. If you do not install the Statistics subcomponent, set STATISTIC=NO. The default is YES.

Note: Record X'4001' is not filterable and X'47' is filterable only by specifying both ACCOUNT=NO and STATISTIC=NO.

STOP=*yyyy–mm–dd–hh.mm.ss.t*
The transaction date and time stopping point for processing, where *yyyy–mm–dd–hh.mm.ss.t* is the year, month, day, hour, minute, second, and tenth of a second timestamp. The default is the end of the log.

TABLEFLUSH=*nnnn*
Specifies how many seconds an entry can remain in log procedure internal tables. The log procedure removes entries older than this limit from the tables when the specified or defaulted FLUSHTYPE conditions are met.

For example, if you specify TABLEFLUSH=5 and flush processing is invoked, transaction entries that started more than 5 seconds before the date, and time found in the last processed relevant record, are removed from the tables.

If you do not specify FLUSHTYPE or a TABLEFLUSH value, table entries accumulate until transaction sets are complete, and the tables are not flushed. This can cause excessive virtual storage use and/or poor performance if transaction sets become large, or collect process termination if a short-on-storage condition occurs.

Possible values are non-negative integers. The default is 1 if FLUSHTYPE is specified. There is no default if FLUSHTYPE is not specified.

VERIFY=*xxxx*
Specifies whether the log collector should stop or continue if a mismatch occurs between the log procedure and the IMS log. The default is NONE. When VERIFY is set to WARN or FAIL, the following is checked:

- Attempt to ensure all logs input to the IMS log collector in a single run are for the same release of IMS.
- Attempt to confirm that the version of IMS that a log is produced by is the same as the version specified or defaulted for the collect job.
- Identify all the IMS originating and processing systems participating in the shared queue, and produce a message if the log data set shows characteristics of not being merged properly.

If the VERIFY parameter is set to WARN, messages are produced and processing continues. If the VERIFY parameter is set to FAIL, messages are produced and processing terminates.

If the VERIFY parameter is set (or defaults) to NONE, no verification is performed.

WRITEPENDING=*xxx*

Specifies whether the log procedure writes incomplete table entries to an output record. This can occur when an `X'4001'` record found on the input log file indicates an IMS COLD start or at end-of-job. If `WRITEPENDING=NO` is specified or defaulted, any pending data during COLD start processing is discarded. If the `DRLICHKO` `DDNAME` is present, the log procedure ignores the **`WRITEPENDING`** parameter at end-of-job to avoid the possibility of duplication.

Possible values are `NO` and `YES`. The default is `NO`.

Using DRLSLOGP

DRLSLOGP is a stand-alone batch program that you can use to run the IMS CSQ feature log procedure in a non-Tivoli Decision Support for z/OS environment. DRLSLOGP uses the same log procedure and record procedure as the log collector, except that DB2 is never invoked and DB2 tables are not updated.

You can use DRLSLOGP to extract data from IMS SLDSs in your non-Tivoli Decision Support for z/OS environment and then send the output to the main center where DB2 and Tivoli Decision Support for z/OS are running. You can then populate the tables using either the standard collect (for collecting composite records) or the IMS light feature (for collecting R2 records).

For information on how to build and distribute DRLSLOGP, see “Using the IMS light feature” on page 42.

Input and output data sets

DRLSLOGP uses the same data sets as the log collector described in “Input and output data sets” on page 29. You should also specify one of the following additional output data sets to produce summarized data for subsequent collection into a Tivoli Decision Support for z/OS DB2 database.

Note: As the DRLIRPT2 output is effectively a subset of DRLICOMP, you would normally allocate only one of these DDNAMEs to an output data set.

DRLICOMP — composite record output

The IMS CSQ feature writes composite records containing all the information required to update the Tivoli Decision Support for z/OS DB2 database to the DRLICOMP DDNAME. Allocate DRLICOMP to a variable record length data set with maximum record length 32756 if you want to save the composite records, or allocate it to DD DUMMY if you want to discard them.

| Optionally by specifying DRLIPARM parameter REPORTS=R0(56) you can also collect the transaction level statistics (56FA) records into DRLICOMP. |

DRLIRPT2 — R2 (transaction) record output

The IMS CSQ feature writes R2 records containing only transaction data required to update the Tivoli Decision Support for z/OS DB2 database using the light collect to the DRLIRPT2 DDNAME. Allocate DRLIRPT2 to a fixed record length data set with record length 246 if you want to save the R2 records, or allocate it to DD DUMMY if you want to discard them. To write the R2 records to DRLIRPT2 you must also specify the REPORTS=R2(*nn*) DRLIPARM statement.

Optionally, by specifying REPORTS=R2(56,FF), you can also collect the transaction level statistics (56FA) records. If you do collect the 56FA data, the record length of the DRLIRPT2 data set must be changed to a minimum of 536.

DRLIRPT0

This is an optional data set with a minimum record length of 536. This data set is used in conjunction with the REPORTS=R0(56) DRLIPARM parameter. Using this parameter will cause the transaction level statistics (56FA) records to be collected to the data sets DRLIRPT0 and DRLICOMP. If the DD for DRLIRPT0 is set to DUMMY, the transaction level statistics (56FA) records will be collected to the DRLICOMP data set only. The composite records will continue to be collected to the DRLICOMP data set.

Running DRLSLOGP

About this task

DRLSLOGP reads log records from the input IMS log and invokes the IMS log procedure for each of them. It replicates the log collector functions and maintains the same interface with the log procedure. DRLSLOGP output is written to DRLICOMP and DRLIRPT2.

Two sample jobs that run DRLSLOGP follow. The job in Figure 15 collects a merged or non-shared queue log from DDNAME DRLLOG. The job in Figure 16 on page 41 collects unmerged shared queue logs from DDNAME DRLLOG*n*.

```
//USERIDA   JOB (ACC000,001),'DRLSLOGP',
//          NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//*
//DRLSLOGP  EXEC PGM=DRLSLOGP
//STEPLIB   DD DISP=SHR,DSN=Tivoli Decision Support for z/OS load library
//*
//DRLLOG    DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS
//*
//DRLICOMP  DD DUMMY    -- or LRECL=32756 output composite data set
//DRLICHKI  DD DUMMY    -- or previously created checkpoint data set
//DRLICHKO  DD DUMMY    -- or LRECL=32756 output checkpoint data set
//*
//DRLOUT    DD SYSOUT=*,DCB=(LRECL=80)
//DRLDUMP   DD SYSOUT=*,DCB=(LRECL=32756)
//DRLIRPT2  DD SYSOUT=*,DCB=(LRECL=536)
//DRLIPARM  DD *
IMSVER=B1 -- or IMSVER=C1 (IMS version 12.1)
          -- or IMSVER=D1 (IMS version 13.1)
REPORTS=R2(FF)
//
```

Figure 15. Sample job 1 for running DRLSLOGP

```
//DRLSLOGP JOB (ACC000,001),'DRLSLOGP',
//             NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//DRLSLOGP EXEC PGM=DRLSLOGP
//STEPLIB  DD DISP=SHR,DSN=Tivoli Decision Support for z/OS load library
//DRLLOG1  DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS1
//DRLLOG2  DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS2
//DRLLOG3  DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS3
//DRLLOG4  DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS4
//DRLLOG5  DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS5
//DRLICOMP  DD DUMMY    -- or LRECL=32756 output composite data set
//DRLICHKO  DD DUMMY    -- or previously created checkpoint data set
//DRLICHKI  DD DUMMY    -- or LRECL=32756 output checkpoint data set
//*
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=32756)
//DRLIRPT2 DD SYSOUT=*,DCB=(RECFM=VB,LRECL=536)
//DRLIPARM DD *
SQNLOGS=5
REPORTS=R2(FF)
IMSVER=B1 -- or IMSVER=C1 (IMS version 12.1)
          -- or IMSVER=D1 (IMS version 13.1)
//
```

Figure 16. Sample job 2 for running DRLSLOGP

The DRLLOG DD statement specifies the input IMS log data. If there is a single IMS log, specify the SQNLOGS parameter as 1 or let it default. If you are using logs from different IMS systems as input and you want to merge them internally, then specify DRLLOG1 DD, DRLLOG2 DD, DRLLOG3 DD, ...DRLLOGn DD, specifying one statement for each log.

For example, if you have four IMS systems in the shared-queue environment, and you use the internal log merge, you need to specify four different DRLLOGx DD statements, one for each IMS system, to define its logs:

```
DRLLOG1 DD DISP=SHR,DSN=...
DRLLOG2 DD DISP=SHR,DSN=...
DRLLOG3 DD DISP=SHR,DSN=...
DRLLOG4 DD DISP=SHR,DSN=...
```

and set SQNLOGS=4 in the DRLIPARM.

Note: If you are using only one log, you must specify DRLLOG DD and you cannot specify DRLLOG1 DD.

Specifying DRLSLOGP parameters

You can specify parameters to control the operation of DRLSLOGP. Table 4 on page 42 lists these parameters with a brief description, the default value, and impact on collect performance.

You can specify the parameters in a data set or in the in-stream JCL for DDNAME DRLIPARM. All parameters must start in column 1. There must be an equal sign (=) between the parameter and the value, with no spaces between. No quotation marks, ending colons, or semicolons are allowed. For example, IMSVER for IMS version13, release 1 is specified as `IMSVER=D1`.

Specifying DRLSLOGP parameters

Table 4. Parameter summary for DRLSLOGP

Parameter name	Possible values	Default	Description	Performance impact
IMSVER	B1, C1, D1	D1	IMS version and release number (B1 indicates IMS version 11.1, C1 indicates IMS Version 12.1, and D1 indicates IMS Version 13.1).	None
REPORTS	R2(FF)	None (do not produce R2 records)	Requests record procedure processing and specifies composite record type.	None
	R2(56,FF)	None (do not produce R2 and 56FA records)	Output both R2 and 56FA records to DRLIRPT2.	None
	R0(56)	None (do not produce 56FA records)	Output 56FA records to DRLIRPT0 and DRLICOMP.	None
Note: All parameters in Table 2 on page 34 also apply here because DRLSLOGP runs the log procedure.				

DRLSLOGP parameter descriptions

The parameters for DRLSLOGP are:

IMSVER=*nn*
Specifies which release of IMS log data you are using as input, where *nn* represents the version and release number. For example, if you are using IMS version 13, release 1 data, specify `IMSVER=D1`. D1 is the default.

REPORTS=R2(FF)
Specifies whether the record procedures should be run to produce R2 records. **R2** identifies the record procedure and **FF** identifies the composite record type. The R2 records are written to the ddname DRLIRPT2.

REPORTS=R2(56,FF) or REPORTS=R0(56)
The 56 parameter specifies that the '56FA'x log records are collected to DRLIRPT2 , DRLIRPT0, or DRLICOMP data sets. These records are used for the KMP IMS collect component..

Using the IMS light feature

The IMS light feature collects transaction information only.

By using the DRLSLOGP utility as a stand-alone procedure in the remote location, the R2 records are produced from the complete IMS log. The records (reduced by around 90% compared to the original log size) can then be transferred to the central location where the final collection process occurs, instead of transferring the complete IMS log. This process applies only to the transaction transit time component.

To use this process, read the following sections:

1. “Setting up a load library for the IMS light feature”
2. “Setting up the IMS light feature collect” on page 44

Setting up a load library for the IMS light feature

About this task

You must build and redistribute the load modules library in order to run job DRLSLOGP stand-alone in the remote locations where Tivoli Decision Support for z/OS is not installed. To build this library at the central location where the product is installed, you can customize and use the sample job DRLJCSQL in the product SDRLCNTL library (see the following JCL).

```
//DRLJCSQL JOB (ACCT£), 'IMS SQ LIGHT LIB'
//**********************************************************************
//*                                                                    *
//*    LICENSED MATERIALS - PROPERTY OF IBM                            *
//*                                                                    *
//*    5698-B06 (C) Copyright IBM CORPORATION 2003, 2015               *
//*    SEE COPYRIGHT INSTRUCTIONS                                      *
//*                                                                    *
//**********************************************************************
//*                                                                    *
//*    NAME:  DRLJCSQL                                                 *
//*    STATUS: Tivoli Decision Support for z/OS 1.8.2                  *
//*                                                                    *
//*    FUNCTION:                                                       *
//*       PREPARE LOAD LIBRARY TO RUN STANDALONE THE DRLSLOGP IMS LOG *
//*       PROCEDURE.                                                   *
//*                                                                    *
//*                                                                    *
//*    NOTES:                                                          *
//*       BEFORE YOU SUBMIT THE JOB:                                   *
//*       - CHECK ALL DATA SET NAMES IN LOWER CASE (INLIB, OUTLIB DD  *
//*         STATEMENTS).                                               *
//*       - CHECK ALL THE PARMATERS CAREFULLY (DISP, VOL NAMES IN      *
//*         INLIB AND OUTLIB DD STATEMENTS).                           *
//*         INLIB CONTAINS THE TDS SMP INSTALLED LIBRARY.              *
//*         OUTLIB CONTAINS THE OUTPUT COPY LOAD LIBRARY               *
//*       - DECIDE WHETHER SMP DISTRIBUTION ON TARGET LIBRARIES ARE    *
//*         SUITABLE TO BE USED AS SOURCE AND UPDATE INLIB DSN         *
//*         ACCORDINGLY (SDRLLOAD OR ADRLLOAD)                         *
//*       - DELETE THE SELECT STATEMENT ROW CORRESPONDING TO THE       *
//*         IMS RELEASE YOU ARE NOT INTERESTED IN, IF ANY:             *
//*         DRLSIB1L,... FOR IMS v11r1m0                               *
//*         DRLSIC1L,... FOR IMS v12r1m0                               *
//*         DRLSID1L,... FOR IMS v13r1m0                               *
//*       - DOUBLE CHECK IN THE JOB OUTPUT THAT ALL THE NEEDED         *
//*         LOAD MODULES HAVE BEEN CORRECTLY COPIED.                   *
//**********************************************************************
//STEP1     EXEC PGM=IEBCOPY
//SYSPRINT  DD SYSOUT=A
//INLIB     DD DISP=(SHR,KEEP) ,UNIT=3390, VOL=SER=yyyyy,
//             DSN=drl181.SDRLLOAD
//OUTLIB    DD DSN=drl181.CSQLIGHT.LOAD,DISP=(NEW,CATLG),
//             SPACE=(CYL,(1,1,5)),UNIT=3390,VOL=SER=zzzzzz,
//             DCB=(RECFM=U,BLKSIZE=32760,LRECL=0)
//SYSIN DD *
   COPY I=INLIB,O=OUTLIB
   SELECT MEMBER=(DRLSLOGP, DRL2LOGJ, DRL2CSQJ, DRLPIO24, DRLPMSG)
   SELECT MEMBER=(DRLSIB1L, DRLSIB12, DRLSIB1C)
   SELECT MEMBER=(DRLSIC1L, DRLSIC12, DRLSIC1C)
   SELECT MEMBER=(DRLSID1L, DRLSID12, DRLSID1C)
   /*
```

INLIB The input data set points to the SMP/E installation target load library (SDRLLOAD) from which the required load modules are copied.

OUTLIB
The output data set points to the copy output LOADLIB. It is used in the remote locations as STEPLIB in the DRLSLOGP job.

If you are interested only in a specific version of IMS, the SELECT MEMBER statements related to the other IMS versions (DRLSI*xx*L, DRLSI*xx*2 and DRLSI*xx*C) can be deleted from the job.

Setting up a load library for the IMS light feature

After you have built the CSQLIGHT.LOAD library, distribute it to the remote centers where it is required. Ensure that the STEPLIB in the DRLSLOGP job used in the remote centers is updated accordingly. For example:

```
//DRLILOGP JOB (ACCT#),'DRLSLOGP IMS'
//***************************************************************
//*                                                             *
//*  LICENSED MATERIALS - PROPERTY OF IBM                       *
//*                                                             *
//*  5698-B06 (C) COPYRIGHT IBM CORPORATION 2004, 2015          *
//*  SEE COPYRIGHT INSTRUCTIONS.                                *
//*                                                             *
//***************************************************************
//*                                                             *
//*  NAME: DRLILOGP                                             *
//*                                                             *
//*  STATUS: Tivoli Decision Support for z/OS 1.8.2             *
//*                                                             *
//*  FUNCTION:                                                  *
//*    RUN THE IMS CSQ LOG PROCEDURE STAND ALONE                *
//*                                                             *
//*  NOTES:                                                     *
//*    BEFORE YOU SUBMIT THE JOB:                               *
//*    - CHECK ALL DATA SET NAMES.                              *
//*    - CHECK ALL THE PARAMETERS CAREFULLY.                    *
//*                                                             *
//***************************************************************
//DRLSLOGP EXEC PGM=DRLSLOGP,PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS')
//STEPLIB  DD DISP=SHR,DSN=drlxxx.CSQLIGHT.LOAD
//DRLLOG   DD DISP=SHR,DSN=... INPUT  IMS SLDS LOG
//DRLICOMP DD DISP=SHR,DSN=... OUTPUT COMPOSITE RECORDS OR DUMMY
//DRLICHKI DD DISP=SHR,DSN=... INPUT  IMS CHECKPOINT FILE OR DUMMY
//DRLICHKO DD DISP=SHR,DSN=... OUTPUT IMS CHECKPOINT FILE OR DUMMY
//DRLOUT   DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP  DD SYSOUT=*,DCB=(RECFM=VB,LRECL=32756)
//DRLIRPT2 DD DISP=SHR,DSN=... OUTPUT RECORDS FROM R2 REPORT/PROCEDURE
//DRLIPARM DD *
IMSVER=D            -- THIS IS THE DEFAULT IMS RELEASE TO BE PROCESSED
REPORTS=R2(FF)      -- CALL THE R2 REPORT/PROCEDURE
/*
```

Setting up the IMS light feature collect

About this task

To run the IMS light feature in the central location where Tivoli Decision Support for z/OS is installed, you must have installed the required IMS CSQ feature subcomponents. You can then collect the R2 records produced by job DRLSLOGP by running a collect for one of the following logs:

- COLLECT CSQ_VB11_COLLECL
- COLLECT CSQ_VC11_COLLECL
- COLLECT CSQ_VD11_COLLECL
- COLLECT CSQ_VD11_VB10
- COLLECT CSQ_VD11_VC10
- COLLECT CSQ_VD11_VD10

Using MAXFREE to allow more pending data to be held

You can increase the parameter **MAXFREE** to acquire more virtual storage for storing pending data. However, there are some circumstances in which the quantity of pending data can increase indefinitely.

For example, Fast Path output messages defined with the system definition TERMINAL macro specifying OPTION=FPACK may never produce a type X'5936' terminal output dequeue log record. In this case the associated transaction nodes will accumulate for the duration of the collect. In this scenario you can use the **TABLEFLUSH** parameter to periodically flush incomplete transaction information based on its age.

If you do not use the ddname DRLICHKO to write checkpoint records at the end of the collect, you can use the **WRITEPENDING** parameter to flush the incomplete transactions at the end of the collect. **WRITEPENDING** is the only method of writing out incomplete transactions when a COLD start is encountered in the log file.

Using DRLSLOGP on remote systems

You can use the stand-alone utility DRLSLOGP to process IMS log data sets on remote systems and transfer the DRLICOMP or DRLIRPT2 output to your central location for collect processing. The DRLICOMP data set can be collected using the normal IMS log collect, but the DRLIRPT2 data set must be collected using the IMS light feature log collect. Using either method the amount of data to be transferred can be significantly smaller than the original logs.

You can further reduce the amount of data to be transferred by using TERSE on the remote system, and UNTERSE on the central system. Note that only the transaction transit time subcomponent can be updated using DRLICOMP or DRLIRPT2 input. For more information about using DRLSLOGP and collecting DRLICOMP and DRLIRPT2, see “Using DRLSLOGP” on page 39.

IEC130I issued for SQNLOGS+1

If SQNLOGS=n, where n is between 1 and 9, you will find an informational message in the collect job output:

```
IEC130I DRLLOGx  DD STATEMENT MISSING
```

where x = n + 1.

Using the light feature with HALDB data

About this task

The IMS light feature does not update the IMS_HALDB_OLR_x tables. Therefore, if you are using this feature and you want to collect High Availability Large Database Reorganization data, follow these steps:

1. From the IMS logs that you are processing, extract IMS type 29, subtype 50 records. This could be performed using DFSORT for example.
2. Transfer the R2 data and the IMS type 29 subtype 50 to the z/OS system where Tivoli Decision Support is installed.
3. Run a collect on HALDB OLR records to populate the IMS HALDB OLR tables.
4. Run a collect by using the IMS light feature on the R2 records to populate the other IMS tables.

Using TABLEFLUSH to prevent storage shortage failures

About this task

You can use the **TABLEFLUSH** and **FLUSHTYPE** parameters to prevent collect failures due to short-on-storage conditions. To do this specify FLUSHTYPE=SOS and TABLEFLUSH=0. You can use a larger value for **TABLEFLUSH** but this means the short-on-storage condition is more likely to recur sooner than if you use a low value

If a short-on-storage condition occurs during a collect you should consider increasing the region size of the collect step, or modifying your **TABLEFLUSH** and **FLUSHTYPE** parameters to minimize virtual storage use as described below. The number of times table flush processing was invoked for a short-on-storage condition is shown in the table flush processing statistics report in DRLOUT.

Using TABLEFLUSH to prevent performance degradation

About this task

IMS transactions can sometimes produce log record sequences that are not expected by the log procedure. This can happen when a transaction loops, when I/O errors occur, or when the application has an unusual design. Sometimes this causes the log procedure to store large amounts of transaction data in the internal tables, which can result in poor performance.

To help handle this situation you can set an upper limit for the collect step REGION size based on past normal collects. Allow the log procedure to flush the problem entries by requesting short-on-storage table flush processing by specifying **FLUSHTYPE=SOS** and **TABLEFLUSH=0**. Be aware that setting the REGION size too low may result in unnecessary table flush processing, while setting it too high may delay or prevent the invocation of short-on-storage table flush processing.

If you use this technique you should expect most collect steps to complete without short-on-storage table flush processing being invoked. Successful use of this technique is indicated by short-on-storage table flush processing being invoked a small number of times when a problem log is processed. It should be possible to relate this back to an application problem occurring in the IMS system at the time the log was generated.

The number of times table flush processing was invoked for a short-on-storage condition is shown in the table flush processing statistics report in DRLOUT.

Using TABLEFLUSH to improve performance

About this task

You can use the **TABLEFLUSH** and **FLUSHTYPE** parameters to improve the performance of the collect process. There is a trade-off between performance gains and data lost by the flush process, so you need to decide what balance is desirable and monitor the table flush statistics to determine if you need to adjust your table flush settings.

This technique is only effective if the poor performance occurs because the log being processed produces long chains of internal table entries. For example if your IMS applications make extensive use of program switching and you specify or default the **DRLIPARM SECONDARY** parameter to YES then large sets of table entries are required to hold information from the related log records being processed.

You can specify either **FLUSHTYPE=CHKPT** or **FLUSHTYPE=COUNT=nnnnnnnn** to have table flush processing occur at regular intervals. Use the **TABLEFLUSH** parameter to control how much data is flushed each time the processing is performed. Review the CPU use information for the collect step in system message IEF374I and the average storage freed per table flush in the table flush processing statistics reported in message DRL2084I to assess the effectiveness of your settings.

For some value of **FLUSHTYPE** specific to a log, reducing that value no longer significantly reduces CPU use, but continues to decrease the proportion of completed input PUOWs (transactions) being flushed, and may start to increase the amount of CPU used by the collect step. Similar considerations apply to varying the **TABLEFLUSH** value. The table flush processing statistics reported in message DRL2084I can be used to check the proportion of completed input PUOWs being flushed. Setting **FLUSHTYPE=COUNT=10000** with **TABLEFLUSH=1** is a good starting point for a balance between CPU use and data loss.

Using TABLEFLUSH to minimize virtual storage use

About this task

You can use the **TABLEFLUSH** and **FLUSHTYPE** parameters to minimize virtual storage use. There is a trade-off between virtual storage reduction and data lost by the flush process, so you need to decide what balance is desirable and monitor the table flush statistics to determine if you need to adjust your table flush settings.

This technique is only effective if the log records being processed require large amounts of virtual storage for processing. For example, if your IMS applications make extensive use of program switching and you specify or default the **DRLIPARM SECONDARY** parameter to YES then a large amount of virtual storage can be used to hold information from the related log records being processed.

You also need to consider that if you process very large logs the amount of virtual storage used to buffer type X'07' log records can reduce the relative impact of this technique. You can see how much virtual storage has been used for buffering type X'07' log records by reviewing the SET07BUF row of the Total Allocated (bytes) column in the statistics for NODEs created reported in message DRL2072I.

You can specify either **FLUSHTYPE=CHKPT** or **FLUSHTYPE=COUNT=nnnnnnnn** to have table flush processing occur at regular intervals. Use the **TABLEFLUSH** parameter to control how much data is flushed each time the processing is performed.

Review the virtual storage use information for the collect step in system message IEF374I and the average storage freed per table flush in the table flush processing statistics reported in message DRL2084I to assess the effectiveness of your settings.

For some values of **FLUSHTYPE** specific to a log, reducing that value no longer significantly reduces virtual storage use, but continues to decrease the proportion of completed input PUOWs (transactions) being flushed, and may start to increase the amount of CPU used by the collect step. Similar considerations apply to varying the **TABLEFLUSH** value. The table flush processing statistics reported in message DRL2084I can be used to check the proportion of completed input PUOWs being flushed. Setting FLUSHTYPE=COUNT=10000 with TABLEFLUSH=1 is a good starting point for a balance between virtual storage use and data loss.

Recovering from abends during collect

The IMS CSQ feature log procedure saves information from the input records and combines this information to produce composite records. However,the Tivoli Decision Support for z/OS restart procedure skips input records that were processed before a DB2 commit. To protect against loss of IMS input records when a failure occurs during log procedure processing, the IMS CSQ feature log procedure has a checkpoint facility.

Recovery using the log procedure checkpoint facility

If the output checkpoint data set, DRLICHKO, was available and filled during a collect abend, and if that data set is used as DRLICHKI in the restart, use the recovery procedure described in the *Administration Guide*.

If you are using the IMS CSQ feature log procedure checkpoint facility, take care if an abend or failure occurs during a collect commit. If such a failure occurs, backout and recovery become more complex. Always allocate a sufficiently large (for example, 250 tracks of 3380 or 3390) checkpoint data set to the DRLICHKO DDNAME to avoid space problems.

Recovery without the checkpoint facility

Recovery when the DRLICHKO data set is not available for use as DRLICHKI can be carried out as follows.

- If the IMS CSQ feature DB2 tables were not updated before the failure, rerun the collect job after correcting the cause of the error.
- If the IMS CSQ feature DB2 tables were updated before the failure, restore the tables to the status before the collect job failed. This can be done from DB2 backup copies of the IMS CSQ performance feature tables. Run the DB2 RECOVER utility, using standard DB2 procedures for point-in-time recovery.

After restoring the tables and correcting the cause of the error, rerun the collect job.

For information about DB2 recovery, see the *DB2 Administration Guide: Volume 2* and *Volume 3*.

Additional capabilities

The following information is not stored in the DB2 tables provided with the IMS CSQ feature, but is available for processing into user-defined tables. See the source for the IMS_V*nnn*_R2 record definition for complete information:

- Transaction abend codes and completion codes from X'07' and X'5938' log records
- Program-to-program switch root information about the transaction and terminal that started a sequence of program to program transactions
- MSC and ISC root information about the transaction or message switch that started the sequence of multi-system transactions

Part 2. IMS CSQ feature reference

Chapter 5. Log and record definitions

This section describes:

- IMS CSQ feature log definitions.
- IMS CSQ feature record definitions corresponding to IMS record types.
- Composite and R2 record definitions as created by the IMS CSQ feature log and record procedures.

For more information about log and record definitions, refer to the *Language Guide and Reference* manual.

Log definitions

Log definitions reside in the product system tables. They define each log to the IMS CSQ feature. You must define a log to Tivoli Decision Support for z/OS before any data can be collected. You specify the log definition that you want to use in the COLLECT statement. See the topic "Collect" in the *Language Guide and Reference* for more information about the COLLECT statement.

Depending on which components you install, one or more of these log definitions are installed, where *nnn* refers to the related IMS version:

CSQ_V*nnn*_COLLECT
This log definition is part of the collect components. It works together with the IMS CSQ feature log procedures and record procedures. Use it for standard product data collection by issuing the COLLECT CSQ_V*nnn*_COLLECT statement.

CSQ_V*nnn*_COLLECL
This log definition is part of the collect components. It works without the IMS CSQ feature log and record procedures to process R2 records created by running DRLSLOGP. Use it for light product data collection by issuing the COLLECT CSQ_V*nnn*_COLLECL statement.

IMS_V*nnn*_SLDS
This log definition is part of the log records components. It does not make use of the IMS CSQ feature log procedures and record procedures. You can use it for special applications, such as troubleshooting, debugging, and detailed IMS log analysis. Use it to collect data for these special applications by issuing the COLLECT IMS_V*nnn*_SLDS statement.

Record definitions

Each record in a log belongs to some record type. Record definitions describe each record type to the log collector.

Descriptions of record definitions

Record definition names for IMS follow this general format:

IMS Common prefix that indicates an IMS record definition

V*nnn* Indicates the release number (for example VD signifies IMS version 13, release 1)

record type
Corresponding IMS record type

Table 5 lists IMS record types with the corresponding IMS CSQ feature record definition name and description. It also indicates support for the IMS record type in the last two columns with these abbreviations:

X The record type is supported for this release of IMS.

- The record type is not supported for this release of IMS.

NA The record type is not applicable for this release of IMS.

Table 5. IMS record types and log records component record definitions

Record type	Record definition	Description
X'01'	IMS_V*nnn*_01	Message queue record (message received from a CNT). This record represents the message and its text and control information as it appears on the IMS message queues. The DRRN indicates the message queue type: X'00......' indicates QBLKS X'04......' indicates short message queue X'08......' indicates long message queue The record contains the indicator of the origin of the message, its destination, whether it uses MSC, and so on. Note that IMS writes this record when a terminal or another network attached system receives a message from a CNT (such as MSC/ISC and FES). It may be input to a program (if it is enqueued to an SMB) or it may switch to another CNT (the latter is known as a message switch). The only occasion when the date and time fields represent an approximation of when the event occurred is for the originating message. Be careful when using these fields for program-to-program switches and MSC/ISC/FES activity.
X'02'	IMS_V*nnn*_02	Command log record. This record represents a condensed version of the command entered by the IMS user or program. If the command string requires a record longer than the logical record length of the log, the record is segmented into several type X'02' records.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'03'	IMS_V*nnn*_03	Message queue record (message received from a PSB or IMS). This record represents the message, and its text and control information as it appears on the IMS message queues. The DRRN indicates the message queue type: • X'00......' to indicate QBLKS • X'04......' to indicate short message queue • X'08......' to indicate long message queue The record contains the indicator of the origin of the message, its destination, whether it uses MSC, and so on. Note that IMS writes this record when a message is received from IMS or a PSB (such as the output from a program or a system-generated message) as sent to the master terminal operator (MTO). This record can be input to an SMB (for a program-to-program switch) or sent to a CNT. The INode, sequence number, and date and time fields (MSGINODE, MSGTISEQ, MSGEDATE and MSGETIME) pass to subsequent messages, which associates subsequent messages with the originating message. However, MSGTISEQ is not propagated for MSC. The date and time fields represent an approximation of when the event occurred only for the originating message. Be careful when using these fields for anything but IMS system-generated output. If a program-to-program switch occurs during a conversation, then the conversational message is not placed in the SPA but rather is present in the type X'03' SMB-generated message (a X'13' is not generated).
X'06'	IMS_V*nnn*_06	IMS event accounting record. IMS writes this record when major IMS system events occur. The Accounting Identifier field (ACIDENT) lists the events that cause this record to be written.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'07'	IMS_V*nnn*_07	Program termination accounting record. IMS accounts for all programs scheduled and terminated under its control with this record (termination) and the type X'08' schedule record. Type X'08' and type X'07' records are related to each other by the first 12 bytes of the recovery token, which are unique for the duration of the IMS session. This termination record contains the date and time of program termination and the resources it consumed during its scheduling. Several messages can be processed during the time this record is scheduled (see field DLRMCNT) and several commits (see field DLRTOKNS) can occur. Therefore, the precise amount of dependent region CPU and the number of DL/I calls cannot be calculated for each message or commit. The only way that the amount of dependent region CPU and number of DL/I calls can be apportioned to the message or commit truly responsible is by: • Calculating the mean • Apportioning according to the proportion of processing time for each message in relation to the total program schedule time • Using regression analysis to find the best fit Despite these restrictions, this record does represent an accurate account of TCB time consumed by the programs, as scheduled in a region, when compared to the time captured by SMF and recorded in SMF type 30. Type X'08' and type X'07' records are written for all region types including MPP, BMP, IFP, and WFI.
X'08'	IMS_V*nnn*_08	Program schedule record. IMS accounts for all programs scheduled and terminated under its control with this record (schedule) and by a type X'07' termination record. The type X'08' and type X'07' records are related to each other through the first 12 bytes of the recovery token. These bytes are represented by fields LINTOKNN and LINTOKNQ, which are unique for the duration of the IMS session. This schedule record contains the date and time of program schedule and the region and schedule initiation type. Type X'08' and type X'07' records are written for all region types including MPP, BMP, IFP, and WFI.
X'09'		Sequential buffering statistics.
X'0A07'	IMS_V*nnn*_0A07	CPI-C program termination.
X'0A08'	IMS_V*nnn*_0A08	CPI-C program initialization.
X'0F'		LE DECB record.
X'10'	IMS_V*nnn*_10	Security violation record. This record indicates that IMS detected a security violation, identifies the precise nature of the violation and specifies whether it is terminal or program-related.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'11'	IMS_V*nnn*_11	Start of conversation record. This record represents the start of a transaction's conversation session. The allocation of a CCB and an SPA maintains the continuity between transactions in this conversation.
X'12'	IMS_V*nnn*_12	End of conversation record. This record represents the termination of the conversation that was started and logged by the type X'11' record. This record is linked to the X'11' Start record of conversation record through the node name.
X'13'	IMS_V*nnn*_13	SPA insert record. This record represents the control information and text inserted to the scratch pad area by a previously started conversation. It is linked to the active transaction through the recovery token field. When written to the log, the DC routine packs this data by removing and flagging removed strings of blanks and zeros. If a program-to-program switch occurs during a conversation, the conversational message is not placed in the SPA, but instead is present in the type X'03' SMB-generated message. A X'13' is not generated in such a case.
X'14'		Switched-line disconnect.
X'15'		Switched-line connect.
X'16'	IMS_V*nnn*_16	Sign-on/off record. This record logs the security maintenance utility (SMU) or Resource Access Control Facility (RACF®) user sign-on and sign-off.
X'18'	IMS_V*nnn*_18	Extended checkpoint record. This record logs the details and checkpoint data for an extended checkpoint.
X'20'	IMS_V*nnn*_20	Database open record. This record indicates that a DL/I database was opened and describes various key characteristics of the database and its files.
X'21'	IMS_V*nnn*_21	Database close record. This record indicates that a DL/I database was closed and provides several key details about the database and its files. A database open X'20' record was created earlier.
X'24'	IMS_V*nnn*_24	Database error record. This record indicates that a DL/I database had an error. It details the program and transaction accessing the database at the time of the error, the time the error occurred, and the relative byte address (RBA) and cylinder cylinder head head record (CCHHR) details of the error.
X'25'		EEQE record.
X'26'		I/O toleration buffer.
X'27'		Database extension.
X'28'		Phase 1 DC record.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'30'	IMS_V*nnn*_30	Message queue prefix changed record. This record logs changes made to the message queue record prefix. It links directly to a previously logged message.
X'31'	IMS_V*nnn*_31	Message queue GU record. This record logs the details of a message that is GU'd from the message queue to be sent to its destination. The destination may be an SMB or CNT. The record is present for incoming messages that are processed by a program scheduled in a message processing region, or for outgoing messages that are sent to a network destination. In addition, the record is present for message switches. The timestamp in this record essentially represents the time that the message ceased waiting on the message queue. If the message is sent to a CNT, a type X'36' record follows, ultimately being followed by a type X'33' Free record, regardless of destination.
X'32'	IMS_V*nnn*_32	Message queue reject record. This record is produced when the MSGQ rejects a message because an error occurred, presumably causing an application program abend.
X'33'	IMS_V*nnn*_33	Message queue DRRN free record. This record indicates that DRRNs were freed from the message queues, the message was deleted, and the DRRNs are available for reuse. This record always indicates that the message is no longer needed by IMS .
X'34'	IMS_V*nnn*_34	Message queue cancel record. This record indicates that the message was canceled from the queue and that a subsequent X'35' Enqueue record was not produced.
X'35'	IMS_V*nnn*_35	Message queue enqueue record. This record indicates that the message in the message queue (logged as a type X'01' or X'03' record) has been placed on the queue for processing. If the destination is an SMB, it is usually waiting on the input queue for the PSB to issue the Get Unique. If the destination is a CNT, it is either a message switch or an outbound message and is waiting in the output queue. This record follows the logging of the message as a X'01' or X'03' record, and precedes the X'31' record that indicates the message has been retrieved from the queue for processing or transmission. The timestamp can be carried from the preceding X'01' or X'03'. Because the timestamp from this record may not reflect the real time the message arrived in the system, exercise caution when using it.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'36'	IMS_V*nnn*_36	Message queue dequeue record. This record indicates that the destination CNT received the message, and the message has been dequeued or deleted. This action is prompted by receiving an SNA definite response or exception response. Under certain circumstances, when the message does not request a definite response, the timestamp in the record reflects the date and time of the next input message, and therefore, the estimated network transit time includes user think time.
X'37'	IMS_V*nnn*_37	Message queue syncpoint transfer record. This record indicates that the message transferred to the permanent destination and reflects that a successful commit occurred.
X'38'	IMS_V*nnn*_38	Message queue syncpoint fail record. This record indicates that a syncpoint failure occurred and the message transfer will not occur.
X'39'		Cleanup outqueue release.
X'3A'		DFSQFIX0 free.
X'3B'		DFSQFIX0 invalid message.
X'3C'		DFSQFIX0 validity check.
X'3D'		DFSQFIX0 QBLK altered.
X'3E'		Message chain update.
X'4001'	IMS_V*nnn*_4001	Checkpoint begin record. This record contains system-wide information about IMS, and represents the beginning of an IMS system checkpoint. This record follows the logged buffer and pool statistics record, but represents the notification of the start of the IMS checkpoint process .
X'4002'	IMS_V*nnn*_4002	Checkpoint message queue record. This record contains a checkpoint of all the allocated queue blocks, short message and long message records at the time of the IMS checkpoint.
X'4003'	IMS_V*nnn*_4003	Checkpoint CNT record. This record contains a checkpoint of all the CNTs defined to the IMS system, and their status at the time of the checkpoint.
X'4004'	IMS_V*nnn*_4004	Checkpoint SMB record. This record contains a checkpoint of all the SMBs defined to the IMS system and their status at the time of the IMS checkpoint.
X'4005'	IMS_V*nnn*_4005	Checkpoint CTB record. This record contains a checkpoint of all the CTBs defined to IMS and their status at the time of the IMS checkpoint.
X'4006'	IMS_V*nnn*_4006	Checkpoint DMB record. This record contains a checkpoint of the database manager blocks (DMBs) defined to the IMS system and their status at the time of the IMS checkpoint.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'4007'	IMS_V*nnn*_4007	Checkpoint PSB record. This record contains a checkpoint of all the PSBs defined to the IMS system at the time of the checkpoint.
X'4008'	IMS_V*nnn*_4008	Checkpoint CLB record. This record contains a checkpoint of all the communications line blocks (CLBs) defined to the IMS system at the time of the IMS checkpoint.
X'4009'		Checkpoint CPT.
X'400A'		Checkpoint CPM.
X'400B'		Checkpoint CTM.
X'400C'		Checkpoint CVB.
X'400D'	IMS_V*nnn*_400D	Checkpoint CCB record. This record contains a checkpoint of the CCBs defined in the IMS system and their status at the time of the IMS checkpoint.
X'400E'	IMS_V*nnn*_400E	Checkpoint SPA record. This record contains a checkpoint of the currently allocated SPAs for active conversations.
X'400F'		Checkpoint LCB.
X'4010'		Checkpoint CRB.
X'4011'		Checkpoint TCM.
X'4014'	IMS_V*nnn*_4014	Checkpoint SPA QB record. This record contains a checkpoint of all the SPA queue blocks (QBs) defined in the IMS system and their status at the time of the IMS checkpoint.
X'4015'	IMS_V*nnn*_4015	Checkpoint EQE record. This record contains a checkpoint of all the generated error queue elements (EQEs) at the time of the IMS checkpoint.
X'4020'	IMS_V*nnn*_4020	Checkpoint CIB record. This record contains a checkpoint of all the communications interface blocks (CIBs) defined to the IMS system at the time of the IMS checkpoint.
X'4021'	IMS_V*nnn*_4021	Checkpoint VTCB record. This record contains a checkpoint of all the VTAM terminal control blocks (VTCBs) defined to the IMS system and their status at the time of the IMS checkpoint.
X'4025'		Checkpoint EEQE.
X'4026'		Checkpoint IEEQE/virtual I/O buffer.
X'4027'		In-doubt extended error queue elements (IEEQE).
X'4028'		Error queue elements (EQEL) for RIS.
X'4030'		Checkpoint SID.
X'4031'		Checkpoint RRE.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'4070'	IMS_V*nnn*_4070	Checkpoint MSDB begin record. This record contains system-wide information about IMS MSDBs, such as at the beginning of the IMS checkpoint.
X'4071'	IMS_V*nnn*_4071	Checkpoint MSDB ECNT record. This record contains main storage database (MSDB) ECNT data for the IMS checkpoint.
X'4072'	IMS_V*nnn*_4072	Checkpoint MSDB header record. This record contains the checkpointed MSDB headers as defined to the IMS system and their contents at the time of the IMS checkpoint.
X'4073'	IMS_V*nnn*_4073	Checkpoint MSDB pagefixed record. This record contains the checkpointed pagefixed MSDBs at the time of the IMS checkpoint.
X'4074',	IMS_V*nnn*_4074	Checkpoint MSDB pageable record. This record contains the checkpointed pageable MSDBs at the time of the IMS checkpoint.
X'4079'	IMS_V*nnn*_4079	Checkpoint MSDB end record. This record indicates that the IMS MSDB checkpoint process is now complete for this IMS checkpoint.
X'4080'	IMS_V*nnn*_4080	Checkpoint Fast Path begin record. This record contains system-wide information about IMS Fast Path, such as at the beginning of the IMS checkpoint.
X'4081'	IMS_V*nnn*_4081	Checkpoint Fast Path ECNT record. This record contains the checkpointed Fast Path ECNT data at the time of the IMS checkpoint.
X'4082'	IMS_V*nnn*_4082	Checkpoint Fast Path EMHB record. This record contains a checkpoint of all the allocated expedited message handler blocks at the time of the IMS checkpoint.
X'4083'	IMS_V*nnn*_4083	Checkpoint Fast Path RCTE record. This record contains a checkpoint of all the routing code table entries defined to IMS and their status at the time of the IMS checkpoint.
X'4084'	IMS_V*nnn*_4084	Checkpoint FP DMCB/DMAC.
X'4085'	IMS_V*nnn*_4085	Checkpoint Fast Path MTO buffer record
X'4086'	IMS_V*nnn*_4086	Checkpoint Fast Path DMHR/DEDB record.
X'4087'	IMS_V*nnn*_4087	Checkpoint Fast Path ADSC record.
X'4088'	IMS_V*nnn*_4088	Checkpoint Fast Path IEEQE record.
X'4089'	IMS_V*nnn*_4089	Checkpoint Fast Path end record. This record indicates that the IMS Fast Path checkpoint process is now complete for this IMS checkpoint.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'4098'	IMS_V*nnn*_4098	Checkpoint end blocks record. This record indicates that the IMS simple checkpoint is now complete.
X'4099'	IMS_V*nnn*_4099	Checkpoint end queues record. This record indicates that the IMS dumpq checkpoint is now complete.
X'41'	IMS_V*nnn*_41	Checkpoint batch record. This record indicates that a batch program has issued a checkpoint.
X'42'	IMS_V*nnn*_42	Log buffer control record. This record indicates the status of IMS at log buffer end of volume and switch times.
X'43'	IMS_V*nnn*_43	Log data set control record. This record indicates the status of the IMS OLDS data sets.
X'4502'	IMS_V*nnn*_4502	Queue pool statistics record. This record contains statistics about the use of the message queue pool at the time of the IMS checkpoint.
X'4503'	IMS_V*nnn*_4503	Format buffer pool statistics record. This record contains statistics about the usage of the format buffer pool at the time of the IMS checkpoint.
X'4504'	IMS_V*nnn*_4504	Database buffer pool statistics record. This record contains statistics about the usage of the database buffer pool at the time of the IMS checkpoint.
X'4505'	IMS_V*nnn*_4505	Main pools statistics record. This record contains statistics about the usage of the principal pools at the time of the IMS checkpoint.
X'4506'	IMS_V*nnn*_4506	Scheduling statistics record. This record contains statistics about scheduling conflicts in IMS at the time of the checkpoint.
X'4507'	IMS_V*nnn*_4507	Logger statistics record. This record contains statistics about the logical logger function of IMS at the time of the IMS checkpoint.
X'4508'	IMS_V*nnn*_4508	VSAM subpool statistics record. This record contains statistics about the VSAM subpools at the time of the IMS checkpoint.
X'4509'	IMS_V*nnn*_4509	Program isolation statistics record. This record contains statistics about IMS program isolation and enqueue/dequeue at the time of the IMS checkpoint.
X'450A'	IMS_V*nnn*_450A	Latch statistics record. This record indicates the status of IMS latches at checkpoint time.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'450B'	IMS_V*nnn*_450B	Dispatch storage statistics record. This record indicates the selective dispatching storage pool status at the time of the checkpoint.
X'450C'	IMS_V*nnn*_450C	DFSCBT00 storage statistics record. This record indicates the status of miscellaneous IMS storage pools at the time of the IMS checkpoint.
X'450D'	IMS_V*nnn*_450D	RECANY (receive any) buffer statistics. This record contains statistics about the VTAM receive any (RECANY) buffer usage at the time of the IMS checkpoint.
X'450E'	IMS_V*nnn*_450E	Storage manager statistics record.
X'450F'	IMS_V*nnn*_450F	Dispatch statistics record.
X'45FF'	IMS_V*nnn*_45FF	Statistics.
X'47'	IMS_V*nnn*_47	Active region statistics record. This record contains information about all active regions, including BMP programs, at the time of the IMS checkpoint.
X'48'	IMS_V*nnn*_48	OLDS padding record. This record contains padding and control information for the IMS OLDS.
X'4C'	IMS_V*nnn*_4C	Program/database start/stop record. This record indicates the starting and stopping of program scheduler blocks (PSBs) and database manager blocks (DMBs). It does not carry a timestamp, but given some locality of reference in relation to other records containing reliable timestamps, an approximation of PSB and DMB availability can be made using this record as the start/stop flag.
X'5050'	IMS_V*nnn*_5050	Full function database update undo/redo successful record. This record indicates that the logging of undo and or redo data for a full function database is complete for a database update.
X'5051'	IMS_V*nnn*_5051	Full function database update unsuccessful record. This record indicates that the update action indicated by the previous X'50' record was unsuccessful.
X'5052'	IMS_V*nnn*_5052	Full function database update undo KSDS insert record. This record contains the undo data for a KSDS insert. The presence of a subsequent X'5050' or X'5051' indicates that the action was successful.
X'53'		CI/CA and space manager.
X'55'	IMS_V*nnn*_ 55FE0001	External subsystem DB2 snap in-doubt record. This record indicates that a DB2 external subsystem had to resolve in-doubt structures for a database.
X'56'	IMS_V*nnn*_56	External subsystem record. This record indicates the status of external subsystem connection and commit processing.
X'56FA'	CSQ_V*nnn*_56FA	Transaction Level Statistics This record contains transaction level statistics when the IMS system has been set up to collect these records.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'5901'	IMS_V*nnn*_5901	Fast Path input message. This record indicates the receipt of an input message to the expedited message handler buffer for Fast Path processing.
X'5903'	IMS_V*nnn*_5903	Fast Path output message. This record indicates the placing of an output message into the expedited message handler buffer, after completion of Fast Path processing.
X'5920'	IMS_V*nnn*_5920	Fast Path MSDB change record. This record indicates the changing of an MSDB.
X'5921'	IMS_V*nnn*_5921	Fast Path DEDB area data set open record. This record indicates the opening of a Fast Path DEDB area data set.
X'5922'	IMS_V*nnn*_5922	Fast Path DEDB area data set close record. This record indicates the closing of a Fast Path DEDB area data set.
X'5923'	IMS_V*nnn*_5923	Fast Path DEDB area data set status record. This record indicates the status of a Fast Path DEDB area data set.
X'5924'	IMS_V*nnn*_5924	Fast Path DEDB area data set EQE creation record. This record indicates the creation of an error queue element for a Fast Path DEDB area data set.
X'5936'	IMS_V*nnn*_5936	Fast Path dequeue message record. This record indicates that an expedited message handler message has been sent and successfully received by its destination node.
X'5937'	IMS_V*nnn*_5937	EMH Fast Path syncpoint record. This record indicates that a successful Fast Path syncpoint occurred, indicating that any messages can be transmitted.
X'5938'	IMS_V*nnn*_5938	EMH Fast Path syncpoint failure record. This record indicates that a Fast Path syncpoint failed and that message transmission may not occur.
X'5942'		Fast Path DMHR dequeue.
X'5947'		Fast Path MSSP image copy.
X'5950'	IMS_V*nnn*_5950	Fast Path DEDB database update record. This record indicates that a Fast Path database online update occurred.
X'5951'	IMS_V*nnn*_5951	Fast Path DEDB database update record. This record indicates an update made in a non-recoverable AREA/DEDB.
X'5953'	IMS_V*nnn*_5953	Fast Path DEDB database update (utilities) record. This record indicates that Fast Path database utilities update occurred.
X'5954'	IMS_V*nnn*_5954	Fast Path DEDB database open record. This record indicates the opening of a Fast Path DEDB database.

Table 5. IMS record types and log records component record definitions (continued)

Record type	Record definition	Description
X'5955'	IMS_V*nnn*_5955	Fast Path sequential dependent syncpoint record. This record indicates that a new sequential dependent buffer was obtained during syncpoint processing.
X'5957'	IMS_V*nnn*_5957	Fast Path database DMAC record.
X'5970'	IMS_V*nnn*_5970	Fast Path hot standby MSDB relocation record. This record indicates that an MSDB has been relocated to the XRF hot standby system during takeover.
X'5B'		Buffered Fast Path.
X'5E'		Image capture of SB handler.
X'5F'		DL/I call trace.
X'62'		OSAM error.
X'63'		Allocate/deallocate.
X'64'		Message discarded by MSC.
X'65'		IRSS and SNA restart.
X'66'		3600 standard record.
X'67'	IMS_V*nnn*_67	Subtypes: 00, 01, 03, 06, E0, ED, EE, EF, FB, FC, FD, FF. Communications trace, DMHR on I/O error, and snap trace records. These records contain internal trace information as requested by the systems trace.
X'67FA'	IMS_V*nnn*_67FA	Trace table log record. This record contains the IMS trace table data.
X'69'		3275 switched unauthorized ID.
X'6C'		MSC link connect/disconnect.
X'6D'		XRF hot standby surveillance.
X'6E'		XRF session miscellaneous.
X'70'		Online change.
X'71'		TCF record.
X'7201'	IMS_V*nnn*_7201	ETO user structure dynamically created.
X'7202'	IMS_V*nnn*_7202	ETO user structure dynamically created.
X'7203'	IMS_V*nnn*_7203	ETO user structure modified.
X'7204'	IMS_V*nnn*_7204	CNT added to an ETO user structure.

Comparison of performance programs

Table 6 on page 64 compares the IMS records used by the IMS CSQ feature with those used by similar performance products and programs. The products and programs are abbreviated in the table:

DFSILTA0
IMS Log Transaction Analysis utility

DFSULTA0
IMS Fast Path Log Analysis utility

TDS CSQ
TDS for z/OS IMS CSQ feature

Table 6. Comparison of the IMS CSQ feature with other products for IMS record types

Record	Description	DFSILTA0	DFSULTA0	TDS CSQ
X'01'	Message received from a CNT	X		X
X'03'	Message received from DL/I	X		X
X'07'	Program termination	X		X
X'08'	Program initiation	X		X
X'11'	Start of conversation			X
X'12'	End of conversation			X
X'13'	SPA record			X
X'31'	Message queue GU	X		X
X'32'	Message queue reject	X		X
X'33'	Message queue free	X		X
X'34'	Message cancel	X		X
X'35'	Message queue enqueue	X		X
X'36'	Message queue dequeue	X		X
X'37'	Syncpoint record			X
X'38'	Message after abend			X
X'4001'	IMS checkpoint begin			X
X'4004'	Checkpoint SMB	X		X
X'4098'	IMS checkpoint end	X		X
X'42'	Log buffer control		X	X
X'45'	Statistics records			X
X'47'	Active region			X
X'56'	External subsystem			X
X'56FA'	Transaction Level Statistics			X
X'5901'	Fast Path input		X	X
X'5903'	Fast Path output		X	X
X'5936'	Fast Path dequeue		X	X
X'5937'	Fast Path syncpoint		X	X
X'5938'	Fast Path abend		X	X

Composite and R2 record definitions

Information about the IMS CSQ composite record definitions is useful if you want to use these records outside of the IMS CSQ feature. For example, you can write your own application to use these records. The composite record definitions are as follows:

CSQ_V*nnn*_COMP
Composite transaction records containing data for a group of related transactions. This record is created by the log procedure.

CSQ_V*nnn*_ST*xxxx*
Account, Availability and Statistics record, where *xxxx* is the record type: 4001, 4502, 4503, 4504, 4505, 4506, 4507, 4508, 4509, 450A, 450B, 450C, 450D, 450E, 47, 07, 08, 06, 0A07.

These records are created by the log procedure.

CSQ_V*nnn*_R2
R2 record containing data for one transaction. This record is created by the record procedure which extracts the information from the composite record.

CSQ_V*nnn*_R2_LIGHT
R2 record containing data for one transaction. This is used by the light feature.

Composite record sections

IMS CSQ composite transaction records are composed of all fields from the source IMS records that are used to populate the transaction tables. Each section of the composite record has a different name and prefix. Table 7 explains the sections of the composite record. Each section is identified by a two-character prefix.

Table 7. Composite record sections

Short description	Name	Prefix	Section contents and explanation
Originating Unit of Work	OUOW	OW	Each composite record contains one OUOW section. This section contains common information for all transactions in the composite record. This information is extracted from these IMS record types: **X'01'** Message queue insert **X'03'** M.Q.I. destination an SMB **X'03'** Message queue enqueue **X'5901'** Aoi exit
Processing Unit of Work	PUOW	PW	Each transaction in the composite record is represented by a PUOW section. The PUOW information is extracted from these IMS record types: **X'01'/X'03'** Message queue insert **X'35'** Message queue enqueue **X'31'** Message queue GU **X'36'** Message queue dequeue **X'37'** Message commit/transfer **X'38'** Message failed **X'5901'** EMH input **X'5903'** EMH output **X'5936'** EMH dequeue **X'5937'** Fast Path commit **X'5938'** Fast Path failed

Table 7. Composite record sections (continued)

Short description	Name	Prefix	Section contents and explanation
Program Schedule Block	PSB	PB	Each program in the composite record is represented by a PSB section. Each PSB section is related to one or more PUOW sections in the composite record. The PSB information is extracted from these IMS record types: **X'08'** Program scheduled **X'07'** Program terminated **Note:** One of these records may be absent when an IMS log is processed. This situation is especially likely in the case of WFI regions and IFP regions.

For more information about log and record definitions, see the *Language Guide and Reference* manual.

Chapter 6. Data tables and lookup tables

The product database is a collection of DB2 tables. Each table contains a fixed number of columns. The number of rows in each table varies with time, because of rows added by the collect function and because of database maintenance.

The process of entering data into the DB2 tables consists of several stages. Tivoli Decision Support for z/OS first summarizes the data from the log in one table. It then summarizes the contents of that table into another table, and so on. An *update definition* specifies how data from one source (a record type or table) enters into one target (always a table).

This section describes the data and lookup tables used by the IMS CSQ feature. For an explanation of the naming standard used, see "Naming standard for tables." For information about the relationships between tables and between records and tables, see "IMS CSQ feature object definitions" on page 26.

Naming standard for tables

Names of the IMS CSQ feature tables use this format:
IMS_*content_suffix*

Where:

- *content* is a description (for example, **IMS_TRAN** for transactions by transaction name and user ID).
- *suffix* indicates the summarization level of the data in the table (for example, IMS_TRAN_**H** for IMS transactions by transaction name and user ID summarized by hour).

  A table name can have these summarization-level suffixes:

T	The table holds nonsummarized data (**timestamped** data).
Q	The table holds data summarized by **quarter hour**.
H	The table holds data summarized by **hour**.
D	The table holds data summarized by **day**.
W	The table holds data summarized by **week**.
M	The table holds data summarized by **month**.

  Lookup tables do not have a suffix.

Table descriptions

Each of the data and lookup table descriptions includes information about the table, a description of each of the key columns, and a description of each of the data columns.

Key columns are marked with a K.

Data columns come after the last key column and are sorted in alphabetic order, with any underscores ignored.

Note: Tables with similar contents (that is, tables with the same name but different suffixes) are described under one heading. For example, "IMS_TRAN_H,_D,_W" on page 90 contains information about three similar tables:

```
KPM_IMS_TRAN_H
KPM_IMS_TRAN_D
KPM_IMS_TRAN_W
```

Except for the DATE column, the contents of these tables are identical. Differences in the contents of similar tables are explained in the column descriptions.

The DATE and TIME information is stored in the standard Structured Query Language (SQL) format and displayed in the local format. The DATE column contains the first day of the week for weekly (_W) tables, and the first day of the month for monthly (_M) tables (if any).

Hexadecimal codes in log records are stored as character data in DB2 tables. For example, a 2-byte field X'FFFF' is stored as a 4-byte character string FFFF.

Control tables

The IMS CSQ feature uses the control tables DAY_OF_WEEK and PERIOD_PLAN, which are used by many Tivoli Decision Support for z/OS features. For complete descriptions of these control tables, refer to the *Administration Guide*.

IMS CSQ feature data tables

The data tables for the IMS CSQ collect components are grouped by the following subcomponents and apply to the following versions:

Table 8. IMS CSQ subcomponent data tables

Subcomponent	**Data tables**
Accounting	“IMS_PSB_ACCOUNT_H,_D,_W” on page 73
Availability	“IMS_AVAILABILITY_D,_W” on page 69
Extended Accounting	“IMS_PSB_ACCOUNT2_H,_D,_W, _M” on page 78
Ext System Tran Transit Time	“IMS_SYSTEM_TRAN2_H,_D,_M” on page 86
HALDB OLR	“IMS_HALDB_OLR_T,_H,_D,_W” on page 72
Statistics	“IMS_CHKPT_IOSAM_T” on page 71 “IMS_CHKPT_POOLS_T” on page 71 “IMS_CHKPT_REGION_T” on page 71 “IMS_CHKPT_STATS_T” on page 71 “IMS_CHKPT_VSAM_T” on page 71
System Tran Transit Time	“IMS_SYSTEM_TRAN_H,_D” on page 82
Transaction Level Statistics	“IMS_TRAN_H,_D,_W” on page 90
Transaction Transit Time	“IMS_TRAN_H,_D,_W” on page 90 “IMS_TRAN_QUEUE_Q,_D” on page 95 “IMS_TRAN_QUEUE_QV,_DV” on page 97
Key Performance Metrics	“KPM_IMS_TRAN_H,_D, _W” on page 98

The IMS CSQ feature maintains separate counts of:

- Full function and Fast Path transactions
- Non message-driven BMP programs
- Response times, including their component parts

The IMS CSQ feature also classifies transactions within response-time boundaries that you can customize. So, you can maintain counts of full function and Fast Path transactions that fall into each boundary.

The tables for the transaction and system subcomponents contain information that is:

- Taken from the records produced by log procedure DRLSI*nn*L and record procedure DRLSI*nn*2 (where *nn* identifies the version of IMS).
- Supported by record definitions CSQ_V*nnn*_COMP and CSQ_V*nnn*_R2 (where *nn* identifies the version of IMS).

Consequently, each table contains:

- Summaries of resources consumed (CPU and DL/I, DEDB and MSDB calls)
- Response-time statistics

IMS_AVAILABILITY_D,_W

These tables provide daily and weekly statistics on the availability of IMS subsystems and regions. They contain consolidated data from the IMS_AVAILABILITY_T table.

For more information about the Availability subcomponent, also see page “The collect components” on page 8.

The default retention period for these tables are:

IMS_AVAILABILITY_D
45 days

IMS_AVAILABILITY_W
365 days

Column name		Data type	Description
DATE	**K**	DATE	Date the availability data applies to. For the _W table, this is the date of the first day of the week.
SYSPLEX_NAME	**K**	Char(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
MVS_SYSTEM_ID	**K**	Char(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
IMS_SYSTEM_ID	**K**	Char(8)	The IMS subsystem ID.
RESOURCE_NAME	**K**	Char(18)	Resource Name.
RESOURCE_TYPE	**K**	Char(8)	Resource Type. Possible values are: **IMSSYS** IMS system **IMSREG** IMS region
AVAIL_OBJ_PCT		DECIMAL (4,1)	Availability objective for the resource, in percent. This is from the column AVAIL_OBJ_PCT in the IMS_AVAIL_RESOURCE lookup table. This value should be compared with the actual availability, which is calculated as: 100*UP_IN_SCHEDULE/ SCHEDULE_HOURS

Column name		Data type	Description
MEASURED_HOURS		FLOAT	Number of hours measured.
SCHEDULE_HOURS		FLOAT	Number of hours the resource is scheduled to be up.
STARTS		SMALLINT	Number of times the resource was started.
STARTS_IN_SCHEDULE		SMALLINT	Number of times the resource was started within the schedule.
STOPS		SMALLINT	Number of times the resource was stopped.
STOPS_IN_SCHEDULE		SMALLINT	Number of times the resource was stopped within the schedule.
UP_HOURS		FLOAT	Number of hours the resource was up.
UP_IN_SCHEDULE		FLOAT	Number of hours the resource was up within the schedule.

IMS_AVAILABILITY_T

This table provides detailed availability data about the IMS subsystem and regions. The data comes from different IMS records. It is updated by the IMS_AVAIL_RESOURCE lookup table.

For more information about the Availability subcomponent, see section on “The collect components” on page 8.

The default retention period for this table is 10 days.

Column name		Data type	Description
SYSPLEX_NAME	**K**	Char(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
MVS_SYSTEM_ID	**K**	Char(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
IMS_SYSTEM_ID	**K**	Char(8)	The IMS subsystem ID. This is from the IMS_SYSTEM_ID collect parameter or from the specific field for the different record type collected.
RESOURCE_NAME	**K**	Char(18)	Resource Name. This is from the RESOURCE_TARGET_NM or from RESOURCE_SOURCE_NM into IMS_AVAIL_RESOURCE lookup table.
RESOURCE_TYPE	**K**	Char(8)	Resource Type. Possible values are: **IMSSYS** IMS system **IMSREG** IMS region
INTERVAL_TYPE	**K**	Char(3)	Interval type. Possible values are: ===, |==, ==|, |=|, XXX, |XX, XX|, |X|, and blank, where: **=** Indicates that the resource is up (available) **X** Indicates that the resource is down | Indicates an interval start or end **blank** Means that the status is unknown
START_TIME	**K**	TIMESTAMP	Start time of the interval.
END_TIME		TIMESTAMP	End time of the interval.

Column name		Data type	Description
QUIET_INTERVAL_SEC		INTEGER	Number of seconds after the interval end that the resource is expected to remain in the same status. If another interval with a start time within this range appears, the two interval are merged.

IMS_CHKPT_IOSAM_T

This table contains an unsummarized record of the accumulated counts of ISAM and OSAM buffer pool activity at each IMS system checkpoint.

It relates to the Statistics subcomponent. For more information about this subcomponent, see “Statistics” on page 8.

The default retention period for the table is 7 days.

IMS_CHKPT_POOLS_T

This table contains an unsummarized record of the accumulated and nonaccumulated system pool usage for the I/OP, CWAP, and HIOP pools at each IMS system checkpoint.

It relates to the Statistics subcomponent. For more information about this subcomponent, see page “Statistics” on page 8.

The default retention period for the table is 7 days.

IMS_CHKPT_REGION_T

This table contains an unsummarized record of the dependent regions active at each IMS system checkpoint, and the transactions and programs active at that time, if any.

This table relates to the Statistics subcomponent. For more information about this subcomponent, see page “Statistics” on page 8.

The default retention period for this table is 7 days.

IMS_CHKPT_STATS_T

This table contains an unsummarized record of the accumulated and nonaccumulated IMS system-wide statistics, MSGQ counts, and format buffer pool counts at each IMS system checkpoint.

This table relates to the Statistics subcomponent. For more information about this subcomponent, see “The collect components” on page 8.

The default retention period for the table is 7 days.

IMS_CHKPT_VSAM_T

This table contains an unsummarized record of the accumulated counts of VSAM buffer pool activity at each IMS system checkpoint.

It relates to the Statistics subcomponent. For more information about this subcomponent, see page “Statistics” on page 8.

The default retention period for the table is 7 days.

IMS_HALDB_OLR_T,_H,_D,_W

The IMS_HALDB_OLR_T table provides data on IMS Versions 9 and 10 High Availability Large Database Online Reorganization activity at Unit of Reorganization (UOR) level.

The IMS_HALDB_OLR_H,_D,_W tables provide statistics on IMS Version 9 High Availability Large Database Online Reorganization activity at database partition level.

For IMS Versions 9 and 10 only, these tables relate to the HALDB OLR subcomponent. For more information about this subcomponent, see page "HALDB OLR" on page 8.

All of these tables collect data from IMS type 29, subtype 50 records. The default retention periods are:

IMS_HALDB_OLR_T
7 days

IMS_HALDB_OLR_H
10 days

IMS_HALDB_OLR_D
45 days

IMS_HALDB_OLR_W
365 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activity occurred. From DATE.
TIME	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the **_H** and **_T** tables. From TIME.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function.
PST_ID	**K**	CHAR(2)	The IMS assigned number for the partition specification table (PST) that contains the management and control information for the dependent region that processed the transaction. From PSTNUM.
IMS_ID	**K**	CHAR(8)	IMS system ID. From IMSSYSID.
DBD_NAME	**K**	CHAR(8)	Database description name. From DBDNAME.
PARTITION_NAME	**K**	CHAR(7)	Partition name. It is the database partition where the online reorganization is being done. From PARTNAME.
EXEC_TIME		FLOAT	Execution time, in 0.00 seconds. From HORMTIME.
INPUT_DS		CHAR(5)	Input data sets. Possible value are: A-J, X (X for PHIDAM format, only) and M-V, Y (Y for PHIDAM format, only). From HORMFLG1.
LOCK_COUNT		INTEGER	Count of locks. From HORMLOCK.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.

Column name		Data type	Description
PART_TYPE		CHAR(7)	Partition type. Possible value are PHDAM and PHIDAM. From HORMFLG1.
ROOTS_MOVED		FLOAT	The number of moved roots. From HORMORSZ.
SEG_MOVED		FLOAT	The number of moved segments. From HORMUORS.
SIZE_MOVED		FLOAT	Size moved, in bytes. From HORMUORZ.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
UOR_COUNT		INTEGER	Number of Units of Reorganization (UOR) occurred. This applies only to the **_H, _D, _W** tables.
WAIT_TIME		FLOAT	Wait time, in 0.00 seconds. From HORMWAIT.

IMS_PSB_ACCOUNT_H,_D,_W

These tables contain hourly, daily, and weekly statistics on counts of transactions and resources used by transaction name. They contain information that includes data for transactions scheduling a PSB. The PSB Account tables give statistics for CPU time and elapsed time during a specified period for regions, transactions, and programs (PSB).

These tables can help you determine such things as who is using too much CPU time or, conversely, what programs or transactions, in which regions, are in a wait state too long. From the PGM_CPU_SEC column, you can monitor the actual CPU time required for each transaction. For a given program, the CPU times should be approximately the same across regions and from day to day. However, these mean times should be interpreted based on the number of transactions per scheduling, which is also reported. If the time begins to increase, the most likely reason is increased database activity. This could be a sign that databases need to be reorganized. The other columns in the tables are related to database operations.

The exclusion of child transactions by setting SECONDARY=NO does not affect these tables. They are populated by PSB cumulative data for root and child transactions, even when SECONDARY=NO is specified in the collect job.

You can use these tables to identify transaction utilization and resource consumption on the IMS system.

The default retention periods for these tables are:

IMS_PSB_ACCOUNT_H
10 days

IMS_PSB_ACCOUNT_D
45 days

IMS_PSB_ACCOUNT_W
365 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activities occurred. For the _W table, this is the date of the first day of the week.
TIME	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the _H table.
IMS_SUBSYSTEM_NAME	**K**	CHAR(8)	The IMS subsystem name. From DLRTOKN.

Table descriptions

Column name		Data type	Description
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested. From DLRTRNCD.
PSB_NAME	**K**	CHAR(8)	The name of the IMS program used to process the transaction. This column contains the program specification block (PSB). From DLRNPSB.
PST_ID	**K**	CHAR(2)	The IMS-assigned number for the partition specification table (PST)that contains the management and control information for the dependent region that processed the transaction. The PST can be reused by IMS after a dependent region terminates, so region occupancy and processing analysis are less meaningful if only the region PST ID is used. So, you must also use the region job name (REGION_JOB_NAME) to identify the dependent region.
REGION_JOB_NAME	**K**	CHAR(8)	The MVS- and JES-identified job name for the IMS dependent region. This column uniquely identifies the transaction processing activity for each region, because the region identifier or PST ID can be reused by IMS. From DRLNJOB.
APPLICATION_NAME		CHAR(18)	Application name. This is from the RESOURCE_TARGET_NM in IMS_AVAIL_RESOURCE lookup table. If nothing is found, $UNKNOWN is used as default.
DLI_APSB_CALLS		REAL	Number of DL/I APSB calls, derived from he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAPSB.
DLI_CHKP_CALLS		REAL	Number of DL/I CHKP calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCHKP.
DLI_CMD_CALLS		REAL	The total number of DL/I CMD calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCMD.
DLI_DPSB_CALLS		REAL	Number of DL/I DPSB calls, derived from he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRDPSB.
DLI_EXCL_DEQUEUES		REAL	The total number of DL/I exclusive dequeue calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCDQ.
DLI_EXCL_ENQUEUES		REAL	The total number of DL/I exclusive enqueue calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCNQ.
DLI_EXCL_ENQWAITS		REAL	The total number of waits DL/I-exclusive enqueue calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCWT. This column indicates possible impact due to the degree or type of program isolation activity during this interval, either throughout the system or resulting from this user or transaction.
DLI_GCMD_CALLS		REAL	The total number of DL/I GCMD calls, erived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGCMD.

Column name		Data type	Description
DLI_GMSG_CALLS		REAL	Number of DL/I GMSG calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGMSG.
DLI_ICMD_CALLS		REAL	Number of DL/I ICMD calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRICMD.
DLI_INIT_CALLS		REAL	Number of DL/I INIT calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRINIT.
DLI_INQY_CALLS		REAL	Number of DL/I INQY calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRINQY.
DLI_MSG_AUTH_CALLS		REAL	Number of DL/I message AUTH calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAUTH.
DLI_MSG_CHNG_CALLS		REAL	Number of DL/I message CHNG calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCHNG.
DLI_MSG_SETO_CALLS		REAL	Number of DL/I message SETO calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETO.
DLI_QCMD_DEQUEUES		REAL	The total number of DL/I queue command dequeue calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRQCODQ.
DLI_QCMD_ENQUEUES		REAL	The total number of DL/I queue command enqueue calls, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRQCONQ.
DLI_QCMD_ENQWAITS		REAL	The total number of waits for DL/I queue commands and enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRQCOWT.
DLI_RCMD_CALLS		REAL	Number of DL/I RCMD calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRRCMD.
DLI_ROLB_CALLS		REAL	Number of DL/I ROLB calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRROLB.
DLI_ROLS_CALLS		REAL	Number of DL/I ROLS calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRROLS.
DLI_SETS_CALLS		REAL	Number of DL/I SETS calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETS.
DLI_SETU_CALLS		REAL	Number of DL/I SETU calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETU.
DLI_SLOG_CALLS		REAL	Number of DL/I SLOG calls, derived from he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSLOG.
DLI_TEST_DEQUEUES		REAL	The total number of DL/I test dequeues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRTSTDQ.

Column name		Data type	Description
DLI_TEST_ENQUEUES		REAL	The total number of DL/I test enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRTSTNQ.
DLI_TEST_ENQWAITS		REAL	The total number of DL/I waits on test enqueues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRTSTWT.
DLI_UPDT_DEQUEUES		REAL	The total number of DL/I update dequeues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRSUPDQ.
DLI_UPDT_ENQUEUES		REAL	The total number of DL/I update enqueues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRSUPNQ.
DLI_UPDT_ENQWAITS		REAL	The total number of DL/I waits on update and enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSUPWT.
DLI_XRST_CALLS		REAL	
DLIDB_CALLS		REAL	The total number of DL/I database calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCLCNT.
DLIDB_DEQ_CALLS		FLOAT	
DLIDB_DLET_CALLS		REAL	The total number of DL/I database DLET calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRDLET.
DLIDB_GHN_CALLS		REAL	The total number of DL/I database GHN calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHN.
DLIDB_GHNP_CALLS		REAL	The total number of DL/I database GHNP calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHNP.
DLIDB_GHU_CALLS		REAL	The total number of DL/I database GHU calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHU.
DLIDB_GN_CALLS		REAL	The total number of DL/I database GN calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGN.
DLIDB_GNP_CALLS		REAL	The total number of DL/I database GNP calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGNP.
DLIDB_GU_CALLS		REAL	The total number of DL/I database GU calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGU1.
DLIDB_ISRT_CALLS		REAL	The total number of DL/I database ISRT calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRISRT.
DLIDB_REPL_CALLS		REAL	The total number of DL/I database REPL calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRREPL.
DLIDBCTL_DBIO_SEC		REAL	The total elapsed time for DB I/O for DBCTL, in seconds, derived from the value stored in the program termination record (record type X'07'). Calculated as Sum of DLRTMEIO/1000000.

Column name		Data type	Description
DLIDBCTL_DBIOS		REAL	The total number of DB I/Os for DBCTL, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRIOCNT.
DLIDBCTL_LOCK_SEC		REAL	The total elapsed time for locking for DBCTL, in seconds, derived from the value stored in the program termination record (record type X'07'). Calculated as Sum of DLRTMEPL/1000000.
DLIDC_GN_CALLS		REAL	The total number of DL/I message queue GN calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGUMES.
DLIDC_GU_CALLS		REAL	The total number of DL/I message queue GU calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGUMES.
DLIDC_ISRT_CALLS		REAL	The total number of DL/I message queue ISRT calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRISMES.
DLIDC_PURGE_CALLS		FLOAT	The total number of DL/I message queue PURGE calls derived from the count store.
DLIACC_ACCUM_SEC		REAL	The total transaction time subqueue 6, in seconds, as stored in record type X'07'. This represents the total time spent waiting in a wait-for-input or pseudo wait-for-input region with no work to do. Calculated as Sum of DLRACCQ6.
DLISQ6_ACCUM_SEC		REAL	The Region Subq 6 Time per Message in seconds. Calculated as Sum of DLRSQ6TM.
LAST_MSG_USERID		CHAR(8)	User ID of the last message processed in this dependent region. From DLRUSID.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
PGM_CPU_SEC		REAL	The total dependent region CPU seconds, derived from the count of CPU timer units stored in the program termination record (record type X'07') divided by 38 400 (the number of timer units per CPU second). Calculated as DLRTIME/38400.0 IMS sets the CPU count to 1 timer unit when the **TIMER=** parameter is not used. When this CPU value is found, TDS sets the CPU value to zero.
PGM_FAILURES		REAL	The total number of program abends.
PROGRAM_TYPE		Char(10)	From DRLTYPE. When: **X'80'** UOR END **X'10'** CPI REGION **X'08'** QUICK RESC **X'04'** DBCTL THRD **X'02'** BMP REGION **X'01'** MPP REGION **X'22'** JBP REGION **X'21'** JMP REGION
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.

Column name		Data type	Description
TRANS_PRIOR		CHAR(2)	Transaction priority. From DLRPRTY.
TRANSACTIONS		REAL	Number of transactions processed by the PSB. Calculated as Sum of DLRMCNT. For CPI REGIONs, DBCTL THREADs, and non-message-driven BMPs, this value is a count of the X'07' records.

IMS_PSB_ACCOUNT2_H,_D,_W, _M

These tables contain hourly, daily, weekly, and monthly statistics on counts of transactions and resources used by transaction name and MVS system identifier. They contain information that includes data for transaction scheduling a PSB and give statistics for CPU time and elapsed time during a specified period for Regions, Transactions, Programs (PSB). These tables can help you determine such things as who is using too much CPU time or, conversely, what programs or transactions, in which regions, are in a wait state for too long. From the PGM_CPU_SEC column, you can monitor the actual CPU time required for each transaction. For a given program, the CPU times should be approximately the same across regions and from day to day; however, these mean times should be interpreted based on the number of transactions per scheduling, which is also reported. If the time begins to increase, the most likely reason is increased database activity. This could be a sign that databases need to be reorganized. The other columns in the tables are related to DB operations.

For IMS Versions 9 and 10, these tables relate to the Extended Accounting subcomponent. For more information about this subcomponent, see page “Extended Accounting” on page 8.

The default retention periods for these tables are:

IMS_PSB_ACCOUNT2_H
10 days

IMS_PSB_ACCOUNT2_D
45 days

IMS_PSB_ACCOUNT2_W
365 days

IMS_PSB_ACCOUNT2_M
765 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activities occurred. For the _M table, this is the date of the first day of the month.
TIME	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the _H table.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function.
MVS_SYSTEM_ID	**K**	CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.

Column name		Data type	Description
PSB_NAME	**K**	CHAR(8)	The name of the IMS program used to process the transaction. This column contains the program specification block (PSB).
IMS_SUBSYSTEM_NAME	**K**	CHAR(8)	The IMS subsystem name. From DLRTOKN.
REGION_JOB_NAME	**K**	CHAR(8)	The MVS- and JES-identified job name for the IMS-dependent region. This column uniquely identifies the transaction processing activity for each region, because the region identifier or PST ID can be reused by IMS. From DRLNJOB.
PST_ID	**K**	CHAR(2)	The IMS-assigned number for the partition specification table (PST) that contains the management and control information for the dependent region that processed the transaction. The PST can be reused by IMS after a dependent region terminates, so region occupancy and processing analysis are less meaningful if only the region PST ID is used. So, you must also use the region job name (REGION_JOB_NAME) to identify the dependent region.
APPLICATION_NAME		CHAR(18)	Application name. This is from the RESOURCE_TARGET_NM in IMS_AVAIL_RESOURCE lookup table. If nothing is found, $UNKNOWN is used as default.
DLI_APSB_CALLS		REAL	Number of DL/I APSB calls, derived from he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAPSB.
DLI_CHKP_CALLS		REAL	Number of DL/I CHKP calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCHKP.
DLI_CMD_CALLS		REAL	The total number of DL/I CMD calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCMD.
DLI_DPSB_CALLS		REAL	Number of DL/I DPSB calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRDPSB.
DLI_EXCL_DEQUEUES		REAL	The total number of DL/I exclusive dequeue calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCDQ.
DLI_EXCL_ENQUEUES		REAL	The total number of DL/I exclusive enqueue calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCNQ.
DLI_EXCL_ENQWAITS		REAL	The total number of waits DL/I-exclusive enqueue calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLREXCWT. This column indicates possible impact due to the degree or type of program isolation activity during this interval, either throughout the system or resulting from this user or transaction.
DLI_GCMD_CALLS		REAL	The total number of DL/I GCMD calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGCMD.
DLI_GMSG_CALLS		REAL	Number of DL/I GMSG calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGMSG.

Table descriptions

Column name		Data type	Description
DLI_ICMD_CALLS		REAL	Number of DL/I ICMD calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRICMD.
DLI_INIT_CALLS		REAL	Number of DL/I INIT calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRINIT.
DLI_INQY_CALLS		REAL	Number of DL/I INQY calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRINQY.
DLI_MSG_AUTH_CALLS		REAL	Number of DL/I message AUTH calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAUTH.
DLI_MSG_CHNG_CALLS		REAL	Number of DL/I message CHNG calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCHNG.
DLI_MSG_SETO_CALLS		REAL	Number of DL/I message SETO calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETO.
DLI_PURGE_CALLS		REAL	The total number of DL/I message queue PURGE calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRPUMES.
DLI_QCMD_DEQUEUES		REAL	The total number of DL/I queue command dequeue calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRQCODQ.
DLI_QCMD_ENQUEUES		REAL	The total number of DL/I queue command enqueue calls, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRQCONQ.
DLI_QCMD_ENQWAITS		REAL	The total number of waits for DL/I queue commands and enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRQCOWT.
DLI_RCMD_CALLS		REAL	Number of DL/I RCMD calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRRCMD.
DLI_ROLB_CALLS		REAL	Number of DL/I ROLB calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRROLB.
DLI_ROLS_CALLS		REAL	Number of DL/I ROLS calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRROLS.
DLI_SETS_CALLS		REAL	Number of DL/I SETS calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETS.
DLI_SETU_CALLS		REAL	Number of DL/I SETU calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSETU.
DLI_SLOG_CALLS		REAL	Number of DL/I SLOG calls, derived from he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSLOG.
DLI_TEST_DEQUEUES		REAL	The total number of DL/I test dequeues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRTSTDQ.

Column name		Data type	Description
DLI_TEST_ENQUEUES		REAL	The total number of DL/I test enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRTSTNQ.
DLI_TEST_ENQWAITS		REAL	The total number of DL/I waits on test enqueues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRTSTWT.
DLI_UPDT_DEQUEUES		REAL	The total number of DL/I update dequeues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRSUPDQ.
DLI_UPDT_ENQUEUES		REAL	The total number of DL/I update enqueues, derived from the count stored in the program termination record (record type X'07').Calculated as Sum of DLRSUPNQ.
DLI_UPDT_ENQWAITS		REAL	The total number of DL/I waits on update and enqueues, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSUPWT.
DLI_XRST_CALLS		REAL	
DLIDB_CALLS		REAL	The total number of DL/I database calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRCLCNT.
DLIDB_DEQ_CALLS		FLOAT	
DLIDB_DLET_CALLS		REAL	The total number of DL/I database DLET calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRDLET.
DLIDB_GHN_CALLS		REAL	The total number of DL/I database GHN calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHN.
DLIDB_GHNP_CALLS		REAL	The total number of DL/I database GHNP calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHNP.
DLIDB_GHU_CALLS		REAL	The total number of DL/I database GHU calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGHU.
DLIDB_GN_CALLS		REAL	The total number of DL/I database GN calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGN.
DLIDB_GNP_CALLS		REAL	The total number of DL/I database GNP calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGNP.
DLIDB_GU_CALLS		REAL	The total number of DL/I database GU calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGU1.
DLIDB_ISRT_CALLS		REAL	The total number of DL/I database ISRT calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRISRT.
DLIDB_REPL_CALLS		REAL	The total number of DL/I database REPL calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRREPL.
DLIDBCTL_DBIO_SEC		REAL	The total elapsed time for DB I/O for DBCTL, in seconds, derived from the value stored in the program termination record (record type X'07'). Calculated as Sum of DLRTMEIO/1000000.

Column name		Data type	Description
DLIDBCTL_DBIOS		REAL	The total number of DB I/Os for DBCTL, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRIOCNT.
DLIDBCTL_LOCK_SEC		REAL	The total elapsed time for locking for DBCTL, in seconds, derived from the value stored in the program termination record (record type X'07'). Calculated as Sum of DLRTMEPL/1000000.
DLIDC_GN_CALLS		REAL	The total number of DL/I message queue GN calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGUMES.
DLIDC_GU_CALLS		REAL	The total number of DL/I message queue GU calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRGUMES.
DLIDC_ISRT_CALLS		REAL	The total number of DL/I message queue ISRT calls derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRISMES.
DLIACC_ACCUM_SEC		REAL	The total transaction time subqueue 6, in seconds, as stored in record type X'07'. This represents the total time spent waiting in a wait-for-input or pseudo wait-for-input region with no work to do. Calculated as Sum of DLRACCQ6.
DLISQ6_ACCUM_SEC		REAL	The Region Subq 6 Time per Message in seconds. Calculated as Sum of DLRSQ6TM.
LAST_MSG_USERID		CHAR(8)	User ID of the last message processed in this dependent region. From DLRUSID.
PGM_CPU_SEC		REAL	The total dependent region CPU seconds, derived from the count of CPU timer units stored in the program termination record (record type X'07')divided by 38 400 (the number of timer units per CPU second). Calculated as DLRTIME/38400.0
PGM_FAILURES		REAL	The total number of program abends.
PROGRAM_TYPE		Char(10)	From DRLTYPE. When: **X'80'** UOR END **X'10'** CPI REGION **X'08'** QUICK RESC **X'04'** DBCTL THRD **X'02'** BMP REGION **X'01'** MPP REGION **X'22'** JBP REGION **X'21'** JMP REGION
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANS_PRIOR		CHAR(2)	Transaction priority. From DLRPRTY.
TRANSACTIONS		REAL	Number of transactions processed by the PSB. Calculated as Sum of DLRMCNT. For CPI REGIONs, DBCTL THREADs, and non-message-driven BMPs, this value is a count of the X'07' records.

IMS_SYSTEM_TRAN_H,_D

These tables contain hourly and daily statistics on counts of transactions and response times summarized by IMS system. They contain information that includes data for message-queue-driven transactions and BMPs, EMH-driven Fast Path transactions, and message switches. You can use these tables to view IMS capacity, to monitor service-level trends by system, and for trend analysis of volumes and response times.

For IMS Version 8, these tables relate to the Transaction Transit Time subcomponent. For more information about this subcomponent, see page “Transaction Transit Time” on page 8. For IMS Versions 9 and 10, these tables relate to the System Tran Transit Time subcomponent. For more information about this subcomponent, see page “System Tran Transit Time” on page 8.

The default retention periods for these tables are:

IMS_SYSTEM_TRAN_H
10 days

IMS_SYSTEM_TRAN_D
45 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activities occurred.
TIME	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the **_H** table.
ORIGIN_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.
PROCESS_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.
PST_ID	**K**	CHAR(2)	The IMS-assigned number for the partition specification table (PST)that contains the management and control information for the dependent region that processed the transaction. The PST can be reused by IMS after a dependent region terminates, so region occupancy and processing analysis are less meaningful if only the region PST ID is used. So, you must also use the region job name (REGION_JOB_NAME) to identify the dependent region.
REGION_JOB_NAME	**K**	CHAR(8)	The MVS- and JES-identified job name for the IMS dependent region. This column uniquely identifies the transaction processing activity for each region, because the region identifier or PST ID can be reused by IMS.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function.
FF_ABORTS		REAL	The total number of Full Function transactions that aborted their commits.
FF_COMMITS		REAL	The total number of Full function transactions that completed their commits.
FP_ABORTS		REAL	The total number of Fast Path transactions that aborted their commits.
FP_CI_NHNH_CONT		REAL	The total number of Fast Path CI contentions between non-HSSP and non-HSSP EPCBs, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of CI contentions between non-HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.

Table descriptions

Column name		Data type	Description
FP_COMBINATIONS		REAL	The total number of Fast Path combinations during logging of type X'5950' records, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of combinations during logging of type X'5950' records that the program experienced while the transactions were active.
FP_COMMITS		REAL	The total number of Fast Path transactions that completed their commits.
FP_DEDB_BFR_WAITS		REAL	The total number of Fast Path waits for DEDB buffers, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB buffers that the program experienced while the transactions were active.
FP_DEDB_CALLS		REAL	The total number of Fast Path DEDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB calls that the program experienced while the transactions were active.
FP_DEDB_NH_PUTGET		REAL	The total number of Fast Path DEDB PUTs/GETs on area data sets, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of DEDB PUTs/GETs on area data sets that the program experienced while the transactions were active.
FP_LOGGED_CI		REAL	The total number of Fast Path whole Is logged, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of whole CIs logged by the program while the transactions were active.
FP_MSDB_CALLS		REAL	The total number of Fast Path MSDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of MSDB calls that the program issued while the transactions were active.
FP_OVERFLOW_BFR		REAL	The total number of Fast Path overflow buffers used, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of overflow buffers used by the program while the transactions were active.
INPUT_CSQ		REAL	The total number of input messages issued by transactions and BMP programs queued through IMS CSQ.
INPUT_LOCAL		REAL	The total number of input messages issued by transactions and BMP programs, not using CSQ.
INPUT_SEC		REAL	The total time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
NETWORK_SEC		REAL	The total time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction is so defined in IMS.

Column name		Data type	Description
OUTPUT_CSQ		REAL	The total number of output messages issued by transactions and BMP programs queued through IMS CSQ.
OUTPUT_LOCAL		REAL	The total number of output messages issued by transactions and BMP programs, not using CSQ.
OUTPUT_SEC		REAL	The total time that responding transactions spent on the IMS output queue waiting for transmission to the originating network destination, in seconds.
OUTPUT_CSQ_SEC		REAL	The time between the completed output transaction put on the queue and the get from the queue for routing the output to the terminal. It is always blank for APPC/OTMA transactions.
PGM_SWITCHES		REAL	Number of program-to-program switches, calculated when secondary transactions are processed (SECONDARY=YES).
PGM_SWITCHES_SEC		REAL	The time between the X'07' of the root/children transaction and the X'08' of the children/root transaction. It is calculated when secondary transactions are processed (SECONDARY=YES).
PROCESS_SEC		REAL	The total elapsed time that transactions and BMP programs spent processing in the dependent regions, in seconds.
RESPONSE_SEC		REAL	The total time, in seconds, that responding transactions spent in network transmission to the ultimate destination, as measured using SNA definite response plus host transit time.
RESPONSES		REAL	The total number of responding transactions and BMP programs that sent messages to the originating terminal.
SEC_TRAN_EXCL		REAL	The number of secondary transactions that were not processed (SECONDARY=NO). For each root transaction, this value is calculated as the total number of X'03' records that have MSGQDES flag set to X'81' (destination is an SMB) and are not CSQPUT.
SQ6_TIME		REAL	The total transaction time for subqueue 6, in seconds, as stored in the DL/I GU (record type X'31') and program termination (record type X'07') records. This represents the total time spent waiting in a wait-for- input or pseudo wait-for-input region with no work to do.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRAN_CNTR_1		REAL	The total number of IMS transactions whose transit time was less than the user-specified boundary 1 (default for boundary is 0.7 seconds).
TRAN_CNTR_2		REAL	The total number of IMS transactions whose transit time was less than the user-specified boundary 2 (default for boundary is 1.5 seconds).
TRAN_CNTR_3		REAL	The total number of IMS transactions whose transit time was less than the user-specified boundary 3 (default for boundary is 3 seconds).
TRAN_CNTR_4		REAL	The total number of IMS transactions whose transit time was less than the user-specified boundary 4 (default for boundary is 10 seconds).
TRANSACTIONS		REAL	The total number of IMS transactions for the given interval and unique key combination.

Column name		Data type	Description
TRANSIT_SEC		REAL	The total time, in seconds, transactions and BMP programs spent in the IMS system from first enqueue of the input message to first GU of the responding output message (or transaction termination), excluding the network transmission time.
VSO_NORMAL_BA		REAL	The Normal Buffer Allocation (NBA) value. From IMS record type X'5937'.
VSO_OVER_BA		REAL	The Overflow Buffer Allocation (OBA) value. From IMS record type X'5937'.
VSO_READS_DASD		REAL	Number of reads from DASD to dataspace. From IMS record type X'5937' or X'5938'.
VSO_READS_DTSP		REAL	Number of reads from dataspace. From IMS record type X'5937' or X'5938'.
VSO_UPDS_DTSP		REAL	Number of updates to dataspace. From IMS record type X'5937' or X'5938'.

IMS_SYSTEM_TRAN2_H,_D,_M

Available for IMS Versions 9 and 10, the IMS_SYSTEM_TRAN2_H,_D,_M tables contain hourly, daily, and monthly statistics on counts of transactions and response times. They contain information that include data for message-queue-driven transactions and BMPs, EMH-driven Fast Path transactions, and message switches. Use these tables to identify transaction utilization and subsequent elapsed time, transmission, and queueing effects on the IMS system. Also use them to identify what users did, how their volumes differed, and their response-time experiences.

For IMS Versions 9 and 10, these tables relate to the Extended System Tran Transit Time subcomponent. For more information about this subcomponent, see page "Extended System Transaction Transit Time" on page 8.

The default retention periods are:

IMS_SYSTEM_TRAN2_H
10 days

IMS_SYSTEM_TRAN2_D
45 days

IMS_SYSTEM_TRAN2_M
765 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activities occurred. For the _M table, this is the date of the first day of the month.
TIME (*)	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the **_H** table.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function.
MVS_SYSTEM_ID	**K**	CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.

Column name		Data type	Description
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.
PROGRAM_NAME	**K**	CHAR(8)	The name of the IMS application program used to process the transaction. For full function and Fast Path activity, this column contains the program specification block (PSB) if available. For APPC activity this column contains the TPI used.
TRANSACTION_CLASS	**K**	CHAR(4)	The assigned transaction class.
PERFORMANCE_GROUP	**K**	CHAR(1)	The transaction type based on (DRLMINPUT+ DRLMPROCE) value ranges. It could be: **F = FAST** (0.0 - 0.7) **G = GOOD** (0.7 - 1.5) **M = MEDIUM** (1.5 - 3.0) **L = LOW** (3.0 - 10.0) **S = SLOW** (above 10)
ORIGIN_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.
PROCESS_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.
REGION_JOB_NAME	**K**	CHAR(8)	The MVS- and JES-identified job name for the IMS-dependent region. This column uniquely identifies the transaction processing activity for each region, because the region identifier or PST ID can be reused by IMS.
PST_ID	**K**	CHAR(2)	The IMS-assigned number for the partition specification table (PST) that contains the management and control information for the dependent region that processed the transaction. The PST can be reused by IMS after a dependent region terminates, so region occupancy and processing analysis are less meaningful if only the region PST ID is used. So, you must also use the region job name (REGION_JOB_NAME) to identify the dependent region.
APPC_MODE_NAME		CHAR(8)	Code used by EMH to enable transactions to be routed to programs within LBL. From X'5901' record.
APPC_NETID		CHAR(8)	The destination Network ID for the APPC session.
FF_ABORTS		REAL	The total number of Full Function transactions that aborted their commits.
FF_COMMITS		REAL	The total number of Full function transactions that completed their commits.
FP_ABORTS		REAL	The total number of Fast Path transactions that aborted their commits.
FP_CI_NHNH_CONT		REAL	The total number of Fast Path CI contentions between non-HSSP and non-HSSP EPCBs, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of CI contentions between non-HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.

Column name		Data type	Description
FP_COMBINATIONS		REAL	The total number of Fast Path combinations during logging of type X'5950' records, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of combinations during logging of type X'5950' records that the program experienced while the transactions were active.
FP_COMMITS		REAL	The total number of Fast Path transactions that completed their commits.
FP_DEDB_BFR_WAITS		REAL	The total number of Fast Path waits for DEDB buffers, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB buffers that the program experienced while the transactions were active.
FP_DEDB_CALLS		REAL	The total number of Fast Path DEDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB calls that the program experienced while the transactions were active.
FP_DEDB_NH_PUTGET		REAL	The total number of Fast Path DEDB PUTs/GETs on area data sets, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of DEDB PUTs/GETs on area data sets that the program experienced while the transactions were active.
FP_LOGGED_CI		REAL	The total number of Fast Path whole Is logged, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of whole CIs logged by the program while the transactions were active.
FP_MSDB_CALLS		REAL	The total number of Fast Path MSDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of MSDB calls that the program issued while the transactions were active.
FP_OVERFLOW_BFR		REAL	The total number of Fast Path overflow buffers used, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of overflow buffers used by the program while the transactions were active.
INPUT_CSQ		REAL	The total number of input messages issued by transactions and BMP programs queued through IMS CSQ.
INPUT_LOCAL		REAL	The total number of input messages issued by transactions and BMP programs, not using CSQ.
INPUT_SEC		REAL	The total time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions.
NETWORK_SEC		REAL	The total time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction is so defined in IMS.
OUTPUT_CSQ		REAL	The total number of output messages issued by transactions and BMP programs queued through IMS CSQ.
OUTPUT_LOCAL		REAL	The total number of output messages issued by transactions and BMP programs, not using CSQ.

Column name		Data type	Description
OUTPUT_SEC		REAL	The total time that responding transactions spent on the IMS output queue waiting for transmission to the originating network destination, in seconds.
OUTPUT_CSQ_SEC		REAL	The time between the completed output transaction put on the queue and the get from the queue for routing the output to the terminal. It is always blank for APPC/OTMA transactions.
PGM_CPU_APPROX		REAL	The total dependent region CPU TCB seconds, derived form the count of CPU timer units stored in the program termination record (record type X'07') divided by 38400 (the number of time units per CPU seconds). This column represents the sum of approximate number of CPU seconds of program execution time while the transactions were active. This value is not provided for WFI or PWFI transactions.
PGM_SWITCHES		REAL	Number of program-to-program switches, calculated when secondary transactions are processed (SECONDARY=YES).
PGM_SWITCHES_SEC		REAL	The time between the X'07' of the root/children transaction and the X'08' of the children/root transaction. It is calculated when secondary transactions are processed (SECONDARY=YES).
PROCESS_SEC		REAL	The total elapsed time that transactions and BMP programs spent processing in the dependent regions, in seconds.
RESPONSE_SEC		REAL	The total time, in seconds, that responding transactions spent in network transmission to the ultimate destination, as measured using SNA definite response plus host transit time.
RESPONSES		REAL	The total number of responding transactions and BMP programs that sent messages to the originating terminal.
SEC_TRAN_EXCL		REAL	The number of secondary transactions that were not processed (SECONDARY=NO). For each root transaction, this value is calculated as the total number of X'03' records that have MSGQDES flag set to X'81' (destination is an SMB) and are not CSQPUT.
SQ6_TIME		REAL	The total transaction time for subqueue 6, in seconds, as stored in the DL/I GU (record type X'31') and program termination (record type X'07') records. This represents the total time spent waiting in a wait-for- input or pseudo wait-for-input region with no work to do.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANS_PRIOR		CHAR(2)	Message priority.
TRANSACTIONS		REAL	The total number of IMS transactions for the given interval and unique key combination.
TRANSIT_SEC		REAL	The total time, in seconds, transactions and BMP programs spent in the IMS system from first enqueue of the input message to first GU of the responding output message (or transaction termination), excluding the network transmission time.
VSO_NORMAL_BA		REAL	The Normal Buffer Allocation (NBA) value. From IMS record type X'5937'.
VSO_OVER_BA		REAL	The Overflow Buffer Allocation (OBA) value. From IMS record type X'5937'.

Column name		Data type	Description
VSO_READS_DASD		REAL	Number of reads from DASD to dataspace. From IMS record type X'5937' or X'5938'.
VSO_READS_DTSP		REAL	Number of reads from dataspace. From IMS record type X'5937' or X'5938'.
VSO_UPDS_DTSP		REAL	Number of updates to dataspace. From IMS record type X'5937' or X'5938'.
Note: (*) Field present only in the IMS_SYSTEM_TRAN2_H table.			

IMS_TRAN_H,_D,_W

The IMS_TRAN_H,_D,_W tables contain hourly, daily, and weekly statistics from the IMS Transaction Level Statistics records (56FA).

The IMS_TRAN_H,_D,_W tables contain hourly, daily, and weekly statistics on counts of transactions and response times summarized by transaction name and user ID. They contain information that includes data for message-queue-driven transactions and BMPs, EMH driven Fast Path transactions, and message switches. Use these tables to identify transaction utilization and subsequent elapsed time, transmission, and queuing effects on the IMS system. Use these table to also identify what users did, how their volumes differed, and their response-time experiences.

These tables relate to the Transaction Transit Time subcomponent. For more information about this subcomponent, see page Table 8 on page 68.

The default retention periods are:

IMS_TRAN_H
10 days

IMS_TRAN_D
45 days

IMS_TRAN_W
365 days

Column name		Data type	Description
DATE	**K**	DATE	The date the activities occurred.
TIME	**K**	TIME	The time when the activity started, in the format HH.00.00. This applies only to the **_H** table.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.
PROGRAM_NAME	**K**	CHAR(8)	The name of the IMS application program used to process the transaction. For full function and Fast Path activity, this column contains the program specification block (PSB) if available. For APPC activity this column contains the TPI used.
ROUTING_CODE	**K**	CHAR(8)	Code used by EMH to enable transactions to be routed to programs within LBL. From X'5901' record.
PST_ID	**K**	CHAR(2)	The IMS-assigned number for the partition specification table (PST) that contains the management and control information for the dependent region that processed the transaction. The PST can be reused by IMS after a dependent region terminates, so region occupancy and processing analysis are less meaningful if only the region PST ID is used. So, you must also use the region job name (REGION_JOB_NAME) to identify the dependent region.

Column name		Data type	Description
REGION_JOB_NAME	**K**	CHAR(8)	The MVS- and JES-identified job name for the IMS dependent region. This column uniquely identifies the transaction processing activity for each region, because the region identifier or PST ID can be reused by IMS.
TRANS_TYPE	**K**	CHAR(8)	Activity type as detailed in “TRANS_TYPE key column” on page 94.
PERFORMANCE_GROUP	**K**	CHAR(1)	The transaction type based on (DRLMINPUT+ DRLMPROCE) value ranges. It could be: **F = FAST** (0.0 - 0.7) **G = GOOD** (0.7 - 1.5) **M = MEDIUM** (1.5 - 3.0) **L = LOW** (3.0 - 10.0) **S = SLOW** (above 10)
ORIGIN_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.
PROCESS_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, forexample, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function.
ORIGIN_LTERM	**K**	CHAR(8)	The IMS-defined logical name for the terminal used to request the transaction or OTMA Tpipe name.
DESTINATION_LTERM	**K**	CHAR(8)	The IMS-defined logical name for the terminal used to receive the transaction output. Missing if APPC OTMA.
USER_ID	**K**	CHAR(8)	The user identifier used to gain authorized access to IMS resources. This column contains the logical terminal name if security is not being managed by the IMS-supported /SIGN ON.
APPC_MODE_NAME		CHAR(8)	The mode name for the APPC session.
APPC_NETID		CHAR(8)	The destination Network ID for the APPC session.
FF_ABORTS		REAL	The total number of Full Function transactions that aborted their commits.
FF_COMMITS		REAL	The total number of Full function transactions that completed their commits.
FP_ABORTS		REAL	The total number of Fast Path transactions that aborted their commits.
FP_CI_NHNH_CONT		REAL	The total number of Fast Path CI contentions between non-HSSP and non-HSSP EPCBs, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of CI contentions between non-HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.

Table descriptions

Column name		Data type	Description
FP_COMBINATIONS		REAL	The total number of Fast Path combinations during logging of type X'5950' records, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of combinations during logging of type X'5950' records that the program experienced while the transactions were active.
FP_COMMITS		REAL	The total number of Fast Path transactions that completed their commits.
FP_DEDB_BFR_WAITS		REAL	The total number of Fast Path waits for DEDB buffers, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB buffers that the program experienced while the transactions were active.
FP_DEDB_CALLS		REAL	The total number of Fast Path DEDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of waits for DEDB calls that the program experienced while the transactions were active.
FP_DEDB_NH_PUTGET		REAL	The total number of Fast Path DEDB PUTs/GETs on area data sets, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of DEDB PUTs/GETs on area data sets that the program experienced while the transactions were active.
FP_LOGGED_CI		REAL	The total number of Fast Path whole Is logged, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of whole CIs logged by the program while the transactions were active.
FP_MSDB_CALLS		REAL	The total number of Fast Path MSDB calls, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of MSDB calls that the program issued while the transactions were active.
FP_OVERFLOW_BFR		REAL	The total number of Fast Path overflow buffers used, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of overflow buffers used by the program while the transactions were active.
INPUT_CSQ		REAL	The total number of input messages issued by transactions and BMP programs queued through IMS CSQ.
INPUT_LOCAL		REAL	The total number of input messages issued by transactions and BMP programs, not using CSQ.
INPUT_SEC		REAL	The total time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
NETWORK_SEC		REAL	The total time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction is so defined in IMS.

Column name		Data type	Description
OUTPUT_CSQ		REAL	The total number of output messages issued by transactions and BMP programs queued through IMS CSQ.
OUTPUT_LOCAL		REAL	The total number of output messages issued by transactions and BMP programs, not using CSQ.
OUTPUT_SEC		REAL	The total time that responding transactions spent on the IMS output queue waiting for transmission to the originating network destination, in seconds.
OUTPUT_CSQ_SEC		REAL	The time between the completed output transaction put on the queue and the get from the queue for routing the output to the terminal. It is always blank for APPC/OTMA transactions.
PGM_CPU_APPROX		REAL	The total dependent region CPU TCB seconds, derived form the count of CPU timer units stored in the program termination record (record type X'07') divided by 38400 (the number of time units per CPU seconds). This column represents the sum of approximate number of CPU seconds of program execution time while the transactions were active. This value is not provided for WFI or PWFI transactions.
PGM_SWITCHES		REAL	Number of program-to-program switches, calculated when secondary transactions are processed (SECONDARY=YES).
PGM_SWITCHES_SEC		REAL	The time between the X'07' of the root/children transaction and the X'08' of the children/root transaction. It is calculated when secondary transactions are processed (SECONDARY=YES).
PROCESS_SEC		REAL	The total elapsed time that transactions and BMP programs spent processing in the dependent regions, in seconds.
RESPONSE_SEC		REAL	The total time, in seconds, that responding transactions spent in network transmission to the ultimate destination, as measured using SNA definite response plus host transit time.
RESPONSES		REAL	The total number of responding transactions and BMP programs that sent messages to the originating terminal.
SEC_TRAN_EXCL		REAL	The number of secondary transactions that were not processed (SECONDARY=NO). For each root transaction, this value is calculated as the total number of X'03' records that have MSGQDES flag set to X'81' (destination is an SMB) and are not CSQPUT.
SQ6_TIME		REAL	The total transaction time for subqueue 6, in seconds, as stored in the DL/I GU (record type X'31') and program termination (record type X'07') records. This represents the total time spent waiting in a wait-for- input or pseudo wait-for-input region with no work to do.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANS_PRIOR		CHAR(2)	Message priority.
TRANSACTIONS		REAL	The total number of IMS transactions for the given interval and unique key combination.
TRANSIT_SEC		REAL	The total time, in seconds, transactions and BMP programs spent in the IMS system from first enqueue of the input message to first GU of the responding output message (or transaction termination), excluding the network transmission time.

Column name		Data type	Description
VSO_NORMAL_BA		REAL	The Normal Buffer Allocation (NBA) value. From IMS record type X'5937'.
VSO_OVER_BA		REAL	The Overflow Buffer Allocation (OBA) value. From IMS record type X'5937'.
VSO_READS_DASD		REAL	Number of reads from DASD to dataspace. From IMS record type X'5937' or X'5938'.
VSO_READS_DTSP		REAL	Number of reads from dataspace. From IMS record type X'5937' or X'5938'.
VSO_UPDS_DTSP		REAL	Number of updates to dataspace. From IMS record type X'5937' or X'5938'.

TRANS_TYPE key column

The TRANS_TYPE key column, which is used in the IMS_TRAN_H table, contains transaction characteristics. It also appears in the composite record produced by the R2 record procedure.

Each character in this 8-byte column has a specific meaning:

The first byte is the region type flag:

M	MPP
F	IFP
B	BMP
-	Not available

The second byte is the queue type flag:

C	Common CSQ
L	Local CSQ
-	Not shared

The third byte is the data communication type flag:

M	MSC
I	ISC
A	APPC
C	APPC and MSC, or OTMA and MSC
O	OTMA
-	Not available

The fourth byte is the thread management type flag:

W	WFI or PWFI
Q	Quick reschedule
M	Missing PSB usage values
-	Not available

The fifth byte is the program-to-program switch flag:

P	Primary
S	Secondary
-	Not available

The sixth byte is the mixed mode flag:

F	A transaction starting as Fast Path and ending as full function
P	A transaction starting as full function and ending as Fast Path
-	Not available

The seventh byte is the environment type flag:

J Java™
C CPI-C
- Not available

The eighth byte is the transaction completion status flag:

R Aborted and retried
C Input canceled
A Aborted, or program abended
- Not available

IMS_TRAN_QUEUE_Q,_D

The IMS_TRAN_QUEUE_Q,_D tables provide quarter-hourly and daily statistics on IMS Message Queue usage by IMS transactions.

These tables relate to the Transaction Transit Time subcomponent. For more information about this subcomponent, see page "Transaction Transit Time" on page 8.

The default retention periods for these tables are:

IMS_TRAN_QUEUE_Q
10 days
IMS_TRAN_QUEUE_D
35 days

Column name		Data type	Description
DATE	**K**	DATE	Date when the activity occurred.
TIME	**K**	TIME	The time when the activity started, in the format HH.15.00. This applies only to the _Q table.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 in the weekdays. This column is derived using MVS_SYSTEM_ID, DATE and TIME columns as parameters in the PERIOD function.
ORIGIN_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.
PROCESS_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.
PERFORMANCE_GROUP	**K**	CHAR(1)	The transaction type based on (DRLMINPUT+ DRLMPROCE) value ranges. It could be: **F = FAST** (0.0 - 0.7) **G = GOOD** (0.7 - 1.5) **M = MEDIUM** (1.5 - 3.0) **L = LOW** (3.0 - 10.0) **S = SLOW** (above 10)
QUEUE_TYPE	**K**	CHAR(12)	Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.
INPUT_MSG		REAL	The total number of input messages processed.

Column name		Data type	Description
INPUT_MSG_SEC		REAL	The total time, in seconds, the input messages spent on that queue.
INP_EMHQ_AVG(*)		REAL	Average number of the messages on the input queue for fast path transactions present before processing.
INP_EMHQ_MAX(*)		REAL	Maximum number of the messages on the input queue for fast path transactions present before processing.
INP_EMHQ_MIN(*)		REAL	Minimum number of the messages on the input queue for fast path transactions present before processing.
INP_MSGQ_SHMSG_AVG(*)		REAL	Average number of the messages on the input short message queue for full function transactions present before processing.
INP_MSGQ_SHMSG_MAX(*)		REAL	Maximum number of the messages on the input short message queue for full function transactions present before processing.
INP_MSGQ_SHMSG_MIN(*)		REAL	Minimum number of the messages on the input short message queue for full function transactions present before processing.
INP_MSGQ_LGMSG_AVG(*)		REAL	Average number of the messages on the input long message queue for full function transactions present before processing.
INP_MSGQ_LGMSG_MAX(*)		REAL	Maximum number of the messages on the input long message queue for full function transactions present before processing.
INP_MSGQ_LGMSG_MIN(*)		REAL	Minimum number of the messages on the input long message queue for full function transactions present before processing.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field..
OUTPUT_MSG		REAL	The total number of output messages processed.
OUTPUT_MSG_SEC		REAL	The total time, in seconds, the output messages spent on that queue.
OUT_EMHQ_AVG(*)		REAL	Average number of the messages on the output queue for fast path transactions present before processing.
OUT_EMHQ_MAX(*)		REAL	Maximum number of the messages on the output queue for fast path transactions present before processing.
OUT_EMHQ_MIN(*)		REAL	Minimum number of the messages on the output queue for fast path transactions present before processing.
OUT_MSGQ_SHMSG_AVG(*)		REAL	Average number of the messages on the output short message queue for full function transactions present before processing.
OUT_MSGQ_SHMSG_MAX(*)		REAL	Maximum number of the messages on the output short message queue for full function transactions present before processing.
OUT_MSGQ_SHMSG_MIN(*)		REAL	Minimum number of the messages on the output short message queue for full function transactions present before processing.

Column name		Data type	Description
OUT_MSGQ_LGMSG_AVG(*)		REAL	Average number of the messages on the output long message queue for full function transactions present before processing.
OUT_MSGQ_LGMSG_MAX(*)		REAL	Maximum number of the messages on the output long message queue for full function transactions present before processing.
OUT_MSGQ_LGMSG_MIN(*)		REAL	Minimum number of the messages on the output long message queue for full function transactions present before processing.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANSACTIONS		REAL	The total number of IMS transactions for the given interval and unique key.

Note: The fields marked with (*) are loaded with the following rules:

- The IMS log collected contains an x'4001' record indicating an IMS cold start. The statistics on the queue utilization will start after an IMS cold start has cleared these queues.
- The checkpoint file is necessary to maintain the statistics from one collection to the next. It is also important that the logs do not have gaps between them.
- The rows relative to the Full Function or the Fast Path queue types will load only the relative fields. The other fields will contain a NULL value.
- In the CSQ environment, these fields contain valid data only if the collection is done from a merged log.

IMS_TRAN_QUEUE_QV,_DV

These views provide quarter-hourly and daily statistics on IMS Message Queue usage by IMS transactions. They are based on the IMS_TRAN_QUEUE_Q and _D table.

These views relate to the Transaction Transit Time subcomponent. For more information about the this subcomponent, see page “Transaction Transit Time” on page 8.

Column name		Data type	Description
DATE	**K**	DATE	Date when the activity occurred.
TIME	**K**	TIME	The time when the activity started, in the format HH.15.00. This applies only to the _Q table.
PERIOD_NAME	**K**	CHAR(8)	The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 in the weekdays. This column is derived using MVS_SYSTEM_ID, DATE and TIME columns as parameters in the PERIOD function.
ORIGIN_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.
PROCESS_IMS	**K**	CHAR(8)	The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction the user requested.

Table descriptions

Column name		Data type	Description
PERFORMANCE_GROUP	**K**	CHAR(1)	The transaction type based on (DRLMINPUT+ DRLMPROCE) value ranges. It could be: **F = FAST** (0.0 - 0.7) **G = GOOD** (0.7 - 1.5) **M = MEDIUM** (1.5 - 3.0) **L = LOW** (3.0 - 10.0) **S = SLOW** (above 10)
QUEUE_TYPE	**K**	CHAR(12)	Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.
INPUT_MSG_SEC_AVG		REAL	Average time spent by an input message on that queue. Calculated as INPUT_MSG_SEC / INPUT_MSG.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
OUTPUT_MSG_SEC_AVG		REAL	Average time spent by an output message on that queue. Calculated as OUTPUT_MSG_SEC / OUTPUT_MSG.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
TRANS_RATE		REAL	Average transaction arrival rate over the interval. Calculated as TRANSACTIONS / 900 (for quarter-hourly) or TRANSACTIONS / 86400 (for daily).

Note: In addition to the key and data columns described here, these views also contain all the data columns described in "IMS_TRAN_QUEUE_Q/_D" on page “IMS_TRAN_QUEUE_Q,_D” on page 95.

KPM_IMS_TRAN_H,_D, _W

These tables contain hourly, daily, and weekly statistics from the IMS Transaction Level Statistics records (56FA).

The default retention periods for these tables are:

KPM_IMS_SYSTEM_TRAN_H
10 days

KPM_IMS_SYSTEM_TRAN_D
60 days

KPM_IMS_SYSTEM_TRAN_W
500 days

Column name		Data type	Description
DATE	**K**	DATE	The start date of the UOR (GMT). From the UTC field TPSTARTD.

Column name		Data type	Description
TIME	**K**	TIME	The minute when the activity started. From the UTC field TPSTARTT.
ORIGIN_IMS	**K**	CHAR(8)	The IMS IMS system ID. From TPIMSID.
PSB_NAME	**K**	CHAR(8)	The name of the IMS program used to process the transaction. This column contains the program specification block (PSB). From TPCPPSB.
TRANSACTION_NAME	**K**	CHAR(8)	The name of the IMS transaction user requested. From TPTRA.
REGION TYPE	**K**	CHAR(5)	The type of IMS Region. From TPTYPE.
PROCESS_IMS	**K**	CHAR(8)	The IMS IMS process ID. From TPCPOSSN.
PROGRAM_NAME	**K**	CHAR(8)	Program name. From TPPGMNM.
PST_ID	**K**	CHAR(8)	The IMS assigned number for partition spec table (PST) that contains management and control information for dependent region that processed the transaction. From TPCPSTN.
REGION_JOB_NAME	**K**	CHAR(8)	The MVS and JES identified job for IMS dependent region. This column uniquely identifies the tran processing activity for each region, because the region id or PST ID can be reused by IMS.From TPJOBN.
LTERM_NAME	**K**	CHAR(8)	The name of the logical terminal on which the activity occurred. From TPLTERM.
USER_ID	**K**	CHAR(8)	User ID of the message processed in this dependent region. From TPCPUSID. or if TPCPUSID is blank, zero, or xFF the User ID is set from the value of TPTYPE: x01 = $MPP, X02 = $BMP, x10 = $CPIC-C, x21 = $JMP, x22 = $JBL.
MVS_SYSTEM_ID		CHAR(4)	The MVS (SMF) system ID defined SYS1.PARMLIB (SMFPRMnn) by the systems programmer. This column is derived from the run time parameter **:MVS_SYSTEM_ID** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
SYSPLEX_NAME		CHAR(8)	This column is derived from the run time parameter **:SYSPLEX_NAME** or from the IMS_SYSTEM_NAMES lookup table because IMS log records do not contain this field.
GROUP_NAME		CHAR(8)	Group name for non RACF users, otherwise set to unknown. From TPCPGRPN.
TRANSACTION_CLASS		CHAR(4)	From TPCLASS.
TRANSACTIONS		INTEGER	Transaction count.
CPU_EXEC_TOT		REAL	Sum of CPU execution time for the Unit Of Recovery. From TPEXTUOR.
CPU_EXEC_MIN		REAL	Minimum CPU execution time for the Unit Of Recovery. From TPEXTUOR.
CPU_EXEC_MAX		REAL	Minimum CPU execution time for the Unit Of Recovery. From TPEXTUOR.
ELAPSED_TOT		REAL	TOT of Elapsed Transaction time (microseconds).
ELAPSED_MIN		REAL	Min of Elapsed Transaction time (microseconds). From difference between TPSTARTU AND TPENDU.
ELAPSED_MAX		REAL	Max of Elapsed Transaction time (microseconds). From difference between TPSTARTU AND TPENDU.
ELAPSED_DB_IO_TOT		REAL	Sum of Elapsed Database IO time (microseconds). From TPTDBIO.

Column name		Data type	Description
ELAPSED_DB_IO_MIN		REAL	The Minimum Elapsed Database IO time (microseconds). From TPTDBIO.
ELAPSED_DB_IO_MAX		REAL	The Maximum Elapsed Database IO time (microseconds). From TPTDBIO.
ELAPSED_DB_LK_TOT		REAL	Sum of Elapsed Database locking time (microseconds). From TPTDBP.
ELAPSED_DB_LK_MIN		REAL	The Minimum Database locking time (microseconds). From TPTDBPL.
ELAPSED_DB_LK_MAX		REAL	The Maximum Database locking time (microseconds). From TPTDBPL.
DLIDB_GU_CALLS		INTEGER	The total number of DL/I database GU calls issued. Calculated as Sum of TPDGU.
DLIDB_GN_CALLS		INTEGER	The total number of DL/I database GN calls issued. Calculated as Sum of TPDGN.
DLIDB_GNP_CALLS		INTEGER	The total number of DL/I database GNP calls issued. Calculated as Sum of TPDGNP.
DLIDB_GHU_CALLS		INTEGER	The total number of DL/I database GHU calls issued. Calculated as Sum of TPDGHU.
DLIDB_GHN_CALLS		INTEGER	The total number of DL/I database GHN calls issued. Calculated as Sum of TPDGHN.
DLIDB_GHNP_CALLS		INTEGER	The total number of DL/I database GHNP calls issued. Calculated as Sum of TPDGHNP.
DLIDB_ISRT_CALLS		INTEGER	The total number of DL/I database ISRT calls issued. Calculated as Sum of TPDISRT.
DLIDB_DLET_CALLS		INTEGER	The total number of DL/I database DLET calls issued. Calculated as Sum of TPDDLET.
DLIDB_REPL_CALLS		INTEGER	The total number of DL/I database REPL calls issued. Calculated as Sum of TPDREPL.
DLIDB_CALLS		INTEGER	The total number of DL/I database calls. Calculated as Sum of TPDCMD.
DLI_MSG_GU		INTEGER	The total number of DL/I message queue GU calls. Calculated as Sum of TPMGU.
DLI_MSG_GN		INTEGER	The total number of DL/I message queue GN calls. Calculated as Sum of TPMGN.
DLI_MSG_ISRT		INTEGER	The total number of DL/I message queue insert calls. Calculated as Sum of TPMISRT.
DLI_MSG_PURG		INTEGER	The total number of DL/I message queue PURGE calls. Calculated as Sum of TPMPURG.
DLI_TEST_ENQUEUES		INTEGER	The total number of DL/I test enqueues. Calculated as Sum of TPTSTNQ.
DLI_TEST_ENQWAITS		INTEGER	The total number of DL/I waits on test enqueues. Calculated as Sum of DPTSTWT.
DLI_TEST_DEQUEUES		INTEGER	The total number of DL/I test dequeues. Calculated as Sum of TPTSTDQ.
DLI_Q_ENQUEUES		INTEGER	The total number of DL/I update enqueues. Calculated as Sum of TPQCONQ.
DLI_Q_ENQWAITS		INTEGER	The total number of DL/I waits on update and enqueues. Calculated as Sum of TPQCOWT.

Column name		Data type	Description
DLI_Q_DEQUEUES		INTEGER	The total number of DL/I update dequeues. Calculated as Sum of TPQCODQ.
DLI_UPDT_ENQUEUES		INTEGER	The total number of DL/I update enqueues. Calculated as Sum of TPSUPNQ.
DLI_UPDT_ENQWAITS		INTEGER	The total number of DL/I waits on update and enqueues. Calculated as Sum of TPSUPWT.
DLI_UPDT_DEQUEUES		INTEGER	The total number of DL/I update dequeues. Calculated as Sum of TPSUPDQ.
DLI_EXCL_ENQUEUES		INTEGER	The total number of DL/I exclusive enqueue calls. Calculated as Sum of TPEXCNQ.
DLI_EXCL_ENQWAITS		INTEGER	The total number of waits DL/I-exclusive enqueue calls. Calculated as Sum of TPEXCWT.
DLI_EXCL_DEQUEUES		INTEGER	The total number of DL/I exclusive dequeue calls. Calculated as Sum of TPEXCDQ.
DLI_CMD_CALLS		INTEGER	The total number of DL/I CMD calls. Calculated as Sum of TPMCMD.
DLI_GCMD_CALLS		INTEGER	The total number of DL/I GCMD calls. Calculated as Sum of TPMGCMD.
DLI_MSG_CHNG_CALLS		INTEGER	Number of DL/I message CHNG calls. Calculated as Sum of TPMCHNG.
DLI_MSG_AUTH_CALLS		INTEGER	Number of DL/I message AUTH calls. Calculated as Sum of TPMAUTH
DLI_MSG_SETO_CALLS		INTEGER	Number of DL/I message SETO calls. Calculated as Sum of TPMSETO.
DLI_APSB_CALLS		INTEGER	Number of DL/I APSB calls. Calculated as Sum of TPSAPSB.
DLI_DPSB_CALLS		INTEGER	Number of DL/I DPSB calls. Calculated as Sum of TPSDPSB.
DLI_GMSG_CALLS		INTEGER	Number of DL/I GMSG calls. Calculated as Sum of TPSGMSG.
DLI_ICMD_CALLS		INTEGER	Number of DL/I ICMD calls. Calculated as Sum of TPSICMD.
DLI_RCMD_CALLS		INTEGER	Number of DL/I RCMD calls. Calculated as Sum of TPSRCMD.
DLI_CHKP_CALLS		INTEGER	Number of DL/I CHKP calls. Calculated as Sum of TPSCHKP.
DLI_XRST_CALLS		INTEGER	Number of DL/I XRST calls. Calculated as Sum of TPSXRST.
DLI_ROLB_CALLS		INTEGER	Number of DL/I ROLB calls. Calculated as Sum of TPSROLB.
DLI_ROLS_CALLS		INTEGER	Number of DL/I ROLS calls. Calculated as Sum of TPSROLS.
DLI_SETS_CALLS		INTEGER	Number of DL/I SETS calls. Calculated as Sum of TPSSETS.
DLI_SETU_CALLS		INTEGER	Number of DL/I SETU calls. Calculated as Sum of TPSSETU.
DLI_INIT_CALLS		INTEGER	Number of DL/I INIT calls.Calculated as Sum of TPSINIT.
DLI_INQY_CALLS		INTEGER	Number of DL/I INQY calls. Calculated as Sum of TPSINQY.
DLI_SLOG_CALLS		INTEGER	Number of DL/I SLOG calls. Calculated as Sum of TPSLOG.
DLIDB_DEQ_CALLS		INTEGER	The total number of DL/I database deque calls. Sum of TPDDEQ.
TLS_VSAM_READS		INTEGER	Number of vsam read calls. From sum of TPVSAMR.
TLS_VSAM_WRITE		INTEGER	Number of vsam write calls. From sum of TPVSAMW.
TLS_OSAM_READS		INTEGER	Number of osam read calls. From sum of TPOSAMR.

Column name		Data type	Description
TLS_OSAM_WRITE		INTEGER	Number of osam write calls. From sum of TPOSAMW.
TLS_TOT_VSOSIO		INTEGER	Sum of vsam and osam calls.
TLS_TOT_ESAF		INTEGER	Number of esaf calls. From sum of TPTOTIO.
TLS_FAST_FLD		INTEGER	Number of Field calls. From sum of TPFLD.
TLS_FAST_POS		INTEGER	Number of fastpath position calls. From sum of TPPOS.
TLS_RLSE_CALLS		INTEGER	Number of release calls. From sum of TPRLSE.
TLS_XQERY_SAVE		INTEGER	Number of xquery save calls. From sum of TPXSAVE.
TLS_XQERY_RSTR		INTEGER	Number of xquery restore calls. From sum of TPXRSTR.
TLS_XQERY_COPY		INTEGER	Number of xquery copy calls. From sum of TPXCOPY.
TLS_ICAL_CALLS		INTEGER	Number of ICAL calls. From sum of TPSICAL.
TLS_GUR_CALLS		INTEGER	Number of GUR calls. From sum of TPDGURL (V12+).

Lookup tables

This section describes the lookup table.

IMS_AVAIL_RESOURCE

This lookup table defines the IMS resources that are used for tracking availability. It also contains the schedule names and availability objectives to use for the different resources.

Column name		Data type	Description
SYSPLEX_NAME	**K**	Char(8)	Sysplex name that the resource is associated with. This can contain global search characters.
MVS_SYSTEM_ID	**K**	Char(4)	MVS system ID that the resource is associated with. This can contain global search characters.
IMS_SYSTEM_ID	**K**	Char(8)	IMS subsystem ID that the resource is associated with. This can contain global search characters.
RESOURCE_SOURCE_NM	**K**	Char(8)	Resource source name. This specifies the resource you want to monitor. This can contain global search characters.
RESOURCE_TYPE	**K**	Char(8)	Resource type: **IMSSYS** IMS system **IMSREG** IMS region **IMSPGM** IMS program **IMSTRAN** IMS transaction IMSPGM and IMSTRAN will be used for tracking application.
AVAIL_OBJ_PCT		Decimal (4,1)	Availability objective for the resource, in percentage.
RESOURCE_CHKPOINT		Integer	Resource checkpoint interval in seconds.
RESOURCE_TARGET_NM		Char(18)	Resource target name. This is the final name that will be associated with the resource. For IMSPGM and IMSTRAN resource types it will be used to specify the application name. If nothing is present, $UNKNOWN is used as default.

Column name		Data type	Description
SCHEDULE_NAME		Char(8)	Schedule name to use for the resource. If nothing is specified, STANDARD is used as default.

Example of Table Contents

Following is an example of the Table Contents

```
         MVS      IMS       RESOURCE              RESOURCE                         AVAIL
SYSPLEX  SYSTEM   SYSTEM    SOURCE     RESOURCE   TARGET        RESOURCE SCHEDULE  OBJ
NAME     ID        ID        NM        TYPE        NM           CHKPOINT  NAME     PCT
-------- -------- -------- ---------- ----------- ------------ --------- -------- -------
%         %        %        IMS71A     IMSSYS      ---          3600     STANDARD  95.0
%         %        %        DSWM%      IMSREG      ---          3600     STANDARD  95.0
%         %        %        MPP%       IMSREG      MPP          3600     STANDARD  95.0
%         %        %        PGM%       IMSPGM      APPL1        ---      STANDARD  95.0
%         %        %        TRAN%      IMSTRAN     APPL1        ---      STANDARD  95.0
```

Mapping between table fields and CSQ records

The following sections map the columns of the tables with the CSQ records from which they are derived.

IMS_TRAN_x (H, D, W)

Table 9. IMS_TRAN_x (H, D, W) Description

Column name	Column description	CSQ record fields
DESTINATION_LTERM	The IMS-defined logical name for the terminal used to receive the transaction output. Missing if APPC OTMA.	DRL_TRANT1, DRL_TERM from CSQ_Vxxx_R2
ROUTING_CODE	Code used by EMH to enable transactions to be routed to programs within LBL. From X5901 record.	DRL_TRANT1, DRL_ROUTCD from CSQ_Vxxx_R2
PGM_SWITCHES	Number of program-to-program switches.	DRLPGMSWN from CSQ_Vxxx_R2
PGM_SWITCHES_SEC	It is the time from the x07 of the root/children transaction and the x08 of the children/root transaction.	DRL_TRANT1, DRLPGMSWS from CSQ_Vxxx_R2
SQ6_TIME	The total transaction time for subqueue 6, in seconds, as storedin the DL/I GU (record type X31) and program termination(record type X07) records. This represents the total time spent waiting in a wait-for-input or pseudo wait-for-input region with no work to do.	DRLSQ6TME from CSQ_Vxxx_R2
TRANS_PRIOR	Message priority.	DRL_TXPRI from CSQ_Vxxx_R2

IMS_TRAN_QUEUE_x (Q, D)

Table 10. IMS_TRAN_QUEUE_x (Q, D) Description

Column name	Column description	CSQ record fields
QUEUE_TYPE	Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.	DRL_TRANT1 DRL_TRANT2from CSQ_Vxxx_R2

Table 10. IMS_TRAN_QUEUE_x (Q, D) Description (continued)

Column name	Column description	CSQ record fields
INPUT_MSG	The total number of input processed.	DRL_TXFLOW DRL_TXFLAGfrom CSQ_Vxxx_R2
INPUT_MSG_SEC	The total time, in seconds, the input messages spent on that queue	DRL_TRANT1 DRLMINPUT from CSQ_Vxxx_R2
INP_EMHQ_MAX	Maximum number of the messages on the input queue for fast path transactions present before processing.	DRL_TRANT1 DRLIEMHQMfrom CSQ_Vxxx_R2
INP_EMHQ_MIN	Minimum number of the messages on the input queue for fast path transactions present before processing.	DRL_TRANT1 DRLIEMHQMfrom CSQ_Vxxx_R2
OUTPUT_MSG	The total number of output processed.	DRL_TXFLAGfrom CSQ_Vxxx_R2
OUTPUT_MSG_SEC	The total time, in seconds, the output messages spent on that queue.	DRL_TRANT1 DRLMOUTPUT from CSQ_Vxxx_R2
OUT_EMHQ_AVG	Average number of the messages on the output queue for fast path transactions present before processing	DRL_TRANT1 DRLOEMHQM DRL_TXFLOW from CSQ_Vxxx_R2
OUT_EMHQ_MAX	Maximum number of the messages on the output queue forfast path transactions present before processing	DRL_TRANT1 DRLOEMHQM from CSQ_Vxxx_R2
OUT_EMHQ_MIN	Minimum number of the messages on the output queue for fast path transactions present before processing.	DRL_TRANT1 DRLOEMHQM from CSQ_Vxxx_R2

IMS_SYSTEM_TRAN_x (H, D)

Table 11. IMS_SYSTEM_TRAN_x (H, D) Description

Column name	Column description	CSQ record fields
PGM_SWITCHES	Number of program-to-program switches.	DRLPGMSWN from CSQ_Vxxx_R2
PGM_SWITCHES_SEC	It is the time from the x07 of the root/children transaction and the x08 of the children/root transaction.	DRL_TRANT1, DRLPGMSWS from CSQ_Vxxx_R2
SQ6_TIME	The total transaction time for subqueue 6, in seconds, as storedin the DL/I GU (record type X31) and program termination (record type X07) records. This represents the total time spent waiting in a wait-for- input or pseudo wait-for-input region with no work to do.	DRLSQ6TME from CSQ_Vxxx_R2

IMS_PSB_ACCOUNT_x (H, D, W)

Table 12. IMS_PSB_ACCOUNT_x (H, D, W) Description

Column name	Column description	CSQ record fields
DLI_APSB_CALLS	Number of DL/I APSB calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRAPSB.	DLRAPSB from CSQ_Vxxx_ST07
DLI_CHKP_CALLS	Number of DL/I CHKP calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRCHKP.	DLRCHKP from CSQ_Vxxx_ST07
DLI_DPSB_CALLS	Number of DL/I DPSB calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRDPSB.	DLRDPSB from CSQ_Vxxx_ST07
DLI_GMSG_CALLS	Number of DL/I GMSG calls, derived from the count storedin the program termination record (record type X07).Calculated as Sum of DLRGMSG.	DLRGMSG from CSQ_Vxxx_ST07
DLI_ICMD_CALLS	Number of DL/I ICMD calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRICMD.	DLRICMD from CSQ_Vxxx_ST07
DLI_RCMD_CALLS	Number of DL/I RCMD calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRRCMD.	DLRRCMD from CSQ_Vxxx_ST07
DLI_xRST_CALLS	Number of DL/I XRST calls, derived from the count storedin the program termination record (record type X07).Calculated as Sum of DLRXRST.	DLRXRST from CSQ_Vxxx_ST07
DLI_ROLB_CALLS	Number of DL/I ROLB calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRROLB.	DLRROLB from CSQ_Vxxx_ST07
DLI_ROLS_CALLS	Number of DL/I ROLS calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRROLS.	DLRROLS from CSQ_Vxxx_ST07
DLI_SETS_CALLS	Number of DL/I SETS calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSETS.	DLRSETS from CSQ_Vxxx_ST07
DLI_SETU_CALLS	Number of DL/I SETU calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSETU.	DLRSETU from CSQ_Vxxx_ST07
DLI_INIT_CALLS	Number of DL/I INIT calls, derived from the count stored inthe program termination record (record type X07).Calculated as Sum of DLRINIT.	DLRINIT from CSQ_Vxxx_ST07

Table 12. IMS_PSB_ACCOUNT_x (H, D, W) Description (continued)

Column name	Column description	CSQ record fields
DLI_INQY_CALLS	Number of DL/I INQY calls, derived from the count storedin the program termination record (record type X07). Calculated as Sum of DLRINQY.	DLRINQY from CSQ_Vxxx_ST07
DLI_SLOG_CALLS	Number of DL/I SLOG calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSLOG.	DLRSLOG from CSQ_Vxxx_ST07
TRANS_PRIOR	TRANS_PRIOR CHAR(2) priority.	L0APRTY from CSQ_Vxxx_ST07

Part 3. Appendixes

Appendix A. Reports

Tivoli Decision Support for z/OS produces reports based on the data in the product database. Reports can show data from tables or from views. You can request reports using the product reporting dialog or by submitting batch jobs. Typically, you use online reporting for reports that you use once, and batch reporting for regularly required reports.

This section describes the reports provided with the collect component of the IMS CSQ feature. These reports are intended to be a subset of the reports you use to analyze your IMS activity. They include management, service level, performance, and problem-related reports. These reports are known to be useful in monitoring and analyzing IMS-related activity.

Report format and general description

This section describes the elements that are common among Tivoli Decision Support for z/OS feature reports:

- Report title
- Report ID
- Report group
- Source
- Attributes
- Variables
- Report types
- Standard report formats

Report title

Each report has a title. Each report title begins with an abbreviation that identifies the component. IMS CSQ feature reports begin with **IMS**. The rest of the title describes the report.

Report ID

Each report has a unique report identifier. The report ID consists of:

- The prefix **IMS** or **CSQ**.
- A one-character identifier of the IMS CSQ feature subcomponent that provides the report:

  T Transaction subcomponent report

  Y System subcomponent report

  A Application subcomponent report

  S Statistics subcomponent report

  O HALDB OLR subcomponent report
- Sequential numbers given to the reports in a subcomponent; for example, **IMST03**.

Report group

To make it easier to find reports, Tivoli Decision Support for z/OS organizes reports into report groups, which correspond to feature components. IMS CSQ feature reports belong to the **IMS** and **CSQ** report groups.

Source

Each report contains information adapted from a DB2 table. The table name is listed for each report.

Attributes

Each report has certain attributes associated with it. The attributes enable you to search for reports using the dialog. These attributes are supplied for each report:

- The area the report belongs to (for example IMS, VM, or NETWORK)
- The tasks that the report supports:

  Performance
  Performance control task

  Service
  Service level planning task

  Capacity
  Capacity planning task

  Security
  Security control task

  Configuration
  Configuration management discipline

  Operation
  Operations management discipline

  Change
  Change management discipline

  Problem
  Problem management discipline

These are also specified where appropriate:

- Resource types reported (for example, storage or CPU)
- Performance issue reported (for example, availability or response)
- Presentation forms (for example, trend or overview)
- Time resolution in the report (hourly, daily, weekly, or monthly)

Variables

Each report has several variables associated with it. When you select a report to display, Tivoli Decision Support for z/OS prompts you for the variables listed in the description.

Report types

The IMS CSQ feature produces these types of reports:

Overview
An overview report lists status for all resources of the specified type.

Trend A trend report gives information about the behavior of a resource over a specified period.

Detail (or technical)

A detail report presents detailed information on a selected resource. Use this type of report to get as much information as possible in a critical situation.

Worst case

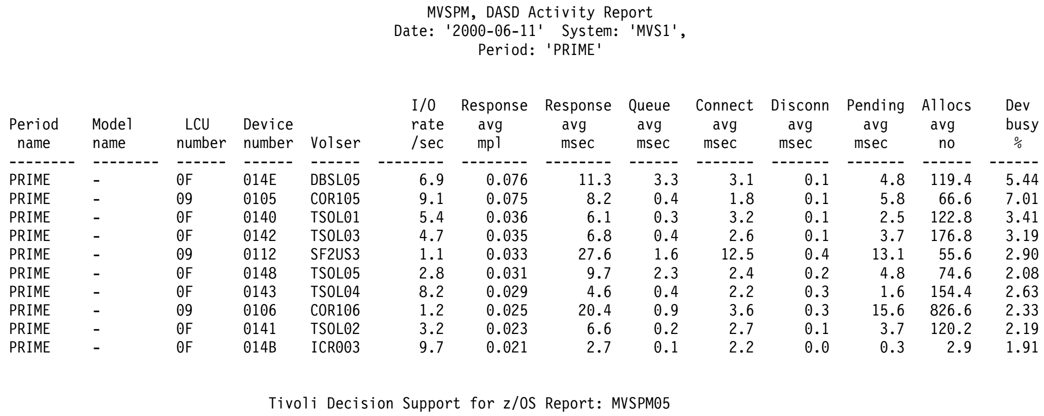
A worst case report lists the resources (usually a maximum of 15) with the worst performance record. However, a worst case report does not imply that the resources listed have a negative performance record, only that the performance has been worse for these than for other resources of the same type.

Standard report formats

Reports are presented in tabular or graphic format. All reports have the same basic report layout. Tabular reports are low-resolution reports that show information in a table format. Graphic reports are high-resolution graphs that give a pictorial representation of the data.

Tabular reports

Figure 17 shows an example of a tabular report.



```
                                MVSPM, DASD Activity Report
                            Date: '2000-06-11'  System: 'MVS1',
                                       Period: 'PRIME'


                                          I/O   Response  Response  Queue   Connect  Disconn  Pending  Allocs    Dev
Period    Model      LCU    Device           rate    avg       avg      avg      avg      avg      avg     avg      busy
 name     name      number  number  Volser   /sec    mpl       msec     msec     msec     msec     msec     no        %
-------- --------  ------  ------  ------  -------- --------  --------  ------  -------  -------  -------  ------  ------
PRIME     -         0F      014E    DBSL05      6.9    0.076      11.3     3.3      3.1      0.1      4.8   119.4    5.44
PRIME     -         09      0105    COR105      9.1    0.075       8.2     0.4      1.8      0.1      5.8    66.6    7.01
PRIME     -         0F      0140    TSOL01      5.4    0.036       6.1     0.3      3.2      0.1      2.5   122.8    3.41
PRIME     -         0F      0142    TSOL03      4.7    0.035       6.8     0.4      2.6      0.1      3.7   176.8    3.19
PRIME     -         09      0112    SF2US3      1.1    0.033      27.6     1.6     12.5      0.4     13.1    55.6    2.90
PRIME     -         0F      0148    TSOL05      2.8    0.031       9.7     2.3      2.4      0.2      4.8    74.6    2.08
PRIME     -         0F      0143    TSOL04      8.2    0.029       4.6     0.4      2.2      0.3      1.6   154.4    2.63
PRIME     -         09      0106    COR106      1.2    0.025      20.4     0.9      3.6      0.3     15.6   826.6    2.33
PRIME     -         0F      0141    TSOL02      3.2    0.023       6.6     0.2      2.7      0.1      3.7   120.2    2.19
PRIME     -         0F      014B    ICR003      9.7    0.021       2.7     0.1      2.2      0.0      0.3     2.9    1.91


                       Tivoli Decision Support for z/OS Report: MVSPM05
```

Figure 17. Tabular reports example

Graphic reports

In some cases, the meaning of data is best presented in graphic form. Graphic reports in Tivoli Decision Support for z/OS have both a QMF format and a Graphical Data Display Manager (GDDM) format. Figure 18 on page 112 illustrates a graphic report.

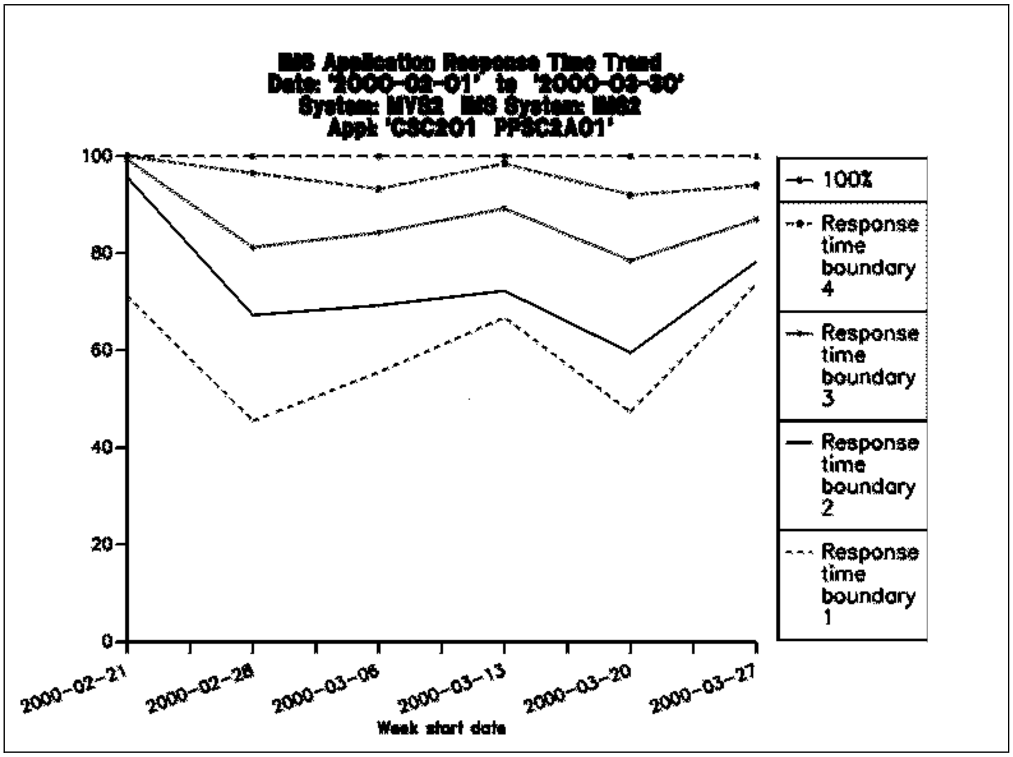


Figure 18. Graphic reports example

For complete information on QMF, refer to the *QMF Learner's Guide* and to the *QMF Advanced User's Guide*.

Samples of reports across non-CSQ and CSQ tables

If you have existing non-CSQ reports that you want to continue to use with the IMS CSQ feature, you may have to create new reports to provide equivalent functionality. This is because the IMS CSQ feature uses tables with different fields and structures compared to the non-CSQ tables. Also, if you wish to report on data from both non-CSQ and IMS CSQ feature tables, then you will need to develop customized reports to combine the data.

Here is a sample using the IMS-shipped report IMSY01: "IMS System Response Time Trend Report". The Query associated with this report is contained in the member DRLQIY01, as shown in Figure 19 on page 113.

```
-----------------------------------------------------------------------
--  Name: DRLQIY01
--
--  Status: Tivoli Decision Support for z/OS
--
--  Function:
--    Define IMS report SQL query.
--
--  CHANGE ACTIVITY:
--
------------------------------------------------------------------------
SELECT
         MVS_SYSTEM_ID
       , IMS_SYSTEM_ID
       , DATE
       , VALUE(
                 ((SUM(EMH_TRAN_CNTR_1) + SUM(MSGQ_TRAN_CNTR_1))
         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
       , VALUE(
         ((SUM(EMH_TRAN_CNTR_2) + SUM(MSGQ_TRAN_CNTR_2))
         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
        , VALUE(
          ((SUM(EMH_TRAN_CNTR_3) + SUM(MSGQ_TRAN_CNTR_3))
          /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
        , VALUE(
          ((SUM(EMH_TRAN_CNTR_4) + SUM(MSGQ_TRAN_CNTR_4))
          /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
FROM &PREFIX.IMS_SYSTEM_D
WHERE
          MVS_SYSTEM_ID   = &MVS_SYSTEM_ID
      AND IMS_SYSTEM_ID   = &IMS_SYSTEM_ID
      AND DATE           >= &FROM_DATE
      AND DATE           <= &TO_DATE
GROUP BY
       MVS_SYSTEM_ID
      , IMS_SYSTEM_ID
      , DATE
```

Figure 19. IMSY01 Report Query (DRLQIY01 member)

This query uses the following fields from the IMS_SYSTEM_D non-CSQ table:

- MVS_SYSTEM_ID
- IMS_SYSTEM_ID
- EMH_TRAN_CNTR_x 1=< x =< 4
- EMH_TRANSACTIONS
- MSGQ_TRAN_CNTR_x 1=< x =< 4
- MSGQ_TRANSACTIONS

Once you have set up the new CSQ implementation engine in the product, you need to use a different query, but with a similar structure to that shown in Figure 20 on page 114:

```
SELECT
                     MVS_SYSTEM_ID
                    ,ORIGIN_IMS
                    ,DATE
                    ,VALUE(
                      (SUM(TRAN_CNTR_1))
                      /(SUM(TRANSACTIONS)), 0)*100
                     ,VALUE(
                      (SUM(TRAN_CNTR_2))
                      /(SUM(TRANSACTIONS)), 0)*100
                     ,VALUE(
                      (SUM(TRAN_CNTR_3))
                      /(SUM(TRANSACTIONS)), 0)*100
                     ,VALUE(
                      (SUM(TRAN_CNTR_4))
                      /(SUM(TRANSACTIONS)), 0)*100
               FROM &PREFIX.IMS_SYSTEM_TRAN_D
               WHERE
                      MVS_SYSTEM_ID   = &MVS_SYSTEM_ID
                 AND ORIGIN_IMS       = &IMS_SYSTEM_ID
                 AND DATE            >= &NEWSQ_FROM_DATE
               GROUP BY
                       MVS_SYSTEM_ID
                      , ORIGIN_IMS
                      , DATE ;
```

Figure 20. Example of Query from IMS_SYSTEM_TRAN_D CSQ Table

This query uses the following fields from the new IMS_SYSTEM_TRAN_D CSQ table:

- MVS_SYSTEM_ID
- ORIGIN_IMS
- TRAN_CNTR_x 1=< x =< 4
- TRANSACTIONS

If you now want to generate a unique report including data from both the non-CSQ table and the new CSQ table you can build a new query using the UNION SQL keyword.

The query might look like this:

```
SELECT
                        MVS_SYSTEM_ID
                       , IMS_SYSTEM_ID
                       , DATE
                       , VALUE(
                         ((SUM(EMH_TRAN_CNTR_1) + SUM(MSGQ_TRAN_CNTR_1))
                         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
                       , VALUE(
                         ((SUM(EMH_TRAN_CNTR_2) + SUM(MSGQ_TRAN_CNTR_2))
                         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
                       , VALUE(
                         ((SUM(EMH_TRAN_CNTR_3) + SUM(MSGQ_TRAN_CNTR_3))
                         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
                       , VALUE(
                         ((SUM(EMH_TRAN_CNTR_4) + SUM(MSGQ_TRAN_CNTR_4))
                         /(SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
                 FROM &PREFIX.IMS_SYSTEM_D                         - - PRE SQ TABLE
                 WHERE
                         MVS_SYSTEM_ID   = &MVS_SYSTEM_ID
                    AND IMS_SYSTEM_ID    = &IMS_SYSTEM_ID
                    AND DATE            >= &PRESQ_FROM_DATE
                    AND DATE            <= &PRESQ_TO_DATE
                 GROUP BY
                        MVS_SYSTEM_ID
                      , IMS_SYSTEM_ID
                      , DATE
                 UNION
                 SELECT
                      MVS_SYSTEM_ID
                     ,ORIGIN_IMS
                     ,DATE
                     ,VALUE(
                       (SUM(TRAN_CNTR_1))
                       /(SUM(TRANSACTIONS)), 0)*100
                      ,VALUE(
                       (SUM(TRAN_CNTR_2))
                       /(SUM(TRANSACTIONS)), 0)*100
                      ,VALUE(
                       (SUM(TRAN_CNTR_3))
                       /(SUM(TRANSACTIONS)), 0)*100
                      ,VALUE(
                       (SUM(TRAN_CNTR_4))
                       /(SUM(TRANSACTIONS)), 0)*100
                 FROM &PREFIX.IMS_SYSTEM_TRAN_D                 - - PRE SQ TABLE
                 WHERE
                       MVS_SYSTEM_ID   = &MVS_SYSTEM_ID
                   AND ORIGIN_IMS      = &IMS_SYSTEM_ID
                   AND DATE          >= &NEWSQ_FROM_DATE
                   AND DATE          <= &NEWSQ_TO_DATE
                 GROUP BY
                        MVS_SYSTEM_ID
                      , ORIGIN_IMS
                      , DATE ;
```

Figure 21. Example of Mixed Query

In Figure 21 you have PRESQ_FROM_DATE and PRESQ_TO_DATE variables used to set the time range of the data needed in the report, from the pre-CSQ environment tables, and NEWSQ_FROM_DATE and NEWSQ_TO_DATE variables used to set the time range of the data needed in the report from the new CSQ environment tables.

For example, assuming that you want to run this report for the whole month of January 2015, but until January, 14th you used the non-CSQ feature, and from January, 15th you started using the CSQ feature, when you are prompted for the input variables in the data selection panel (DRLDRSEL), you can select the correct date range, as follows:

```
Variable            Value
MVS_SYSTEM_ID       ZOS1
IMS_SYSTEM_ID       IMS1
PRESQ_FROM_DATE     2015-01-01
PRESQ_TO_DATE       2015-01-14
NEWSQ_FROM_DATE     2015-01-15
NEWSQ_TO_DATE       2015-01-31
```

Figure 22. Example of Query Input Variables Panel DRLDRSEL

In the report output, you will get the complete data for the whole month, for both the two time periods in which you run the old and the new engine.

IMS detail reports

IMS detail reports give details of items for a specified day.

IMS Message Queue Pool Detail by Date report

This report shows the utilization of the message queue pool buffer for the day and time period selected. Values are shown cumulatively at each IMS checkpoint. Figure 23 on page 117 shows an example of a report.

This information identifies the report:

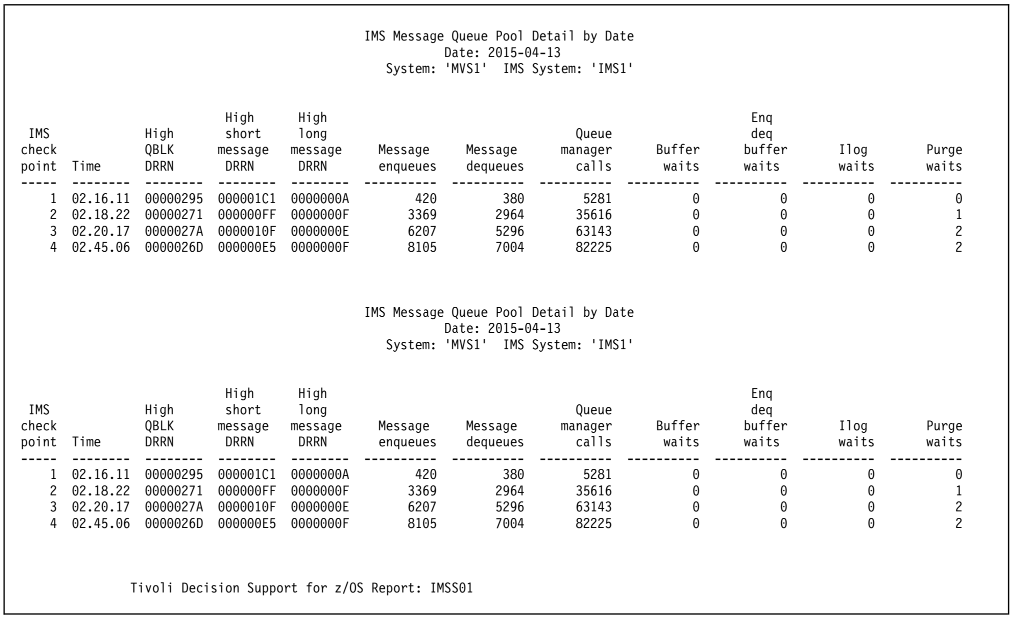
Report ID
IMSS01

Report group
IMS reports

Source
IMS_CHKPT_STATS_T

Attributes
IMS, utilization, performance, qpool, daily

Variables
MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME



```
                                         IMS Message Queue Pool Detail by Date
                                                    Date: 2015-04-13
                                            System: 'MVS1'  IMS System: 'IMS1'


                            High      High                                                         Enq
 IMS            High       short      long                                 Queue                   deq
check           QBLK      message    message     Message     Message     manager      Buffer     buffer        Ilog       Purge
point  Time     DRRN       DRRN       DRRN      enqueues    dequeues       calls       waits      waits        waits       waits
-----  -------- --------  --------  --------  ----------  ----------  ----------  ----------  ----------  ----------  ----------
    1  02.16.11 00000295  000001C1  0000000A         420         380        5281           0           0           0           0
    2  02.18.22 00000271  000000FF  0000000F        3369        2964       35616           0           0           0           1
    3  02.20.17 0000027A  0000010F  0000000E        6207        5296       63143           0           0           0           2
    4  02.45.06 0000026D  000000E5  0000000F        8105        7004       82225           0           0           0           2
```

```
                                         IMS Message Queue Pool Detail by Date
                                                    Date: 2015-04-13
                                            System: 'MVS1'  IMS System: 'IMS1'


                            High      High                                                         Enq
 IMS            High       short      long                                 Queue                   deq
check           QBLK      message    message     Message     Message     manager      Buffer     buffer        Ilog       Purge
point  Time     DRRN       DRRN       DRRN      enqueues    dequeues       calls       waits      waits        waits       waits
-----  -------- --------  --------  --------  ----------  ----------  ----------  ----------  ----------  ----------  ----------
    1  02.16.11 00000295  000001C1  0000000A         420         380        5281           0           0           0           0
    2  02.18.22 00000271  000000FF  0000000F        3369        2964       35616           0           0           0           1
    3  02.20.17 0000027A  0000010F  0000000E        6207        5296       63143           0           0           0           2
    4  02.45.06 0000026D  000000E5  0000000F        8105        7004       82225           0           0           0           2


                Tivoli Decision Support for z/OS Report: IMSS01
```

Figure 23. Example of IMS Message Queue Pool Detail by Date tabular report

The report contains this information:

IMS check point
The numeric ID of the checkpoint for the IMS session.

Time The time when the activity started, in the format HH.MM.SS.

High QBLK DRRN
The DRRN of the highest queue block.

High short message DRRN
The DRRN of the highest short message queue.

High long message DRRN
The DRRN of the highest long message queue.

Message enqueues.
The number of calls to enqueue messages.

Message dequeues.
The number of calls to dequeue messages.

Queue manager calls
The total number of calls to QMGR.

Buffer waits
The number of waits for a free buffer.

Enq deq buffer waits
The number of waits for conflicting enque-dequeue buffer requests.

Ilog waits
The number of waits for ILOG.

Purge waits
The number of waits for purge completion.

IMS OSAM/ISAM Buffer Pool Detail by Date report

This report shows the utilization of IMS OSAM/ISAM buffers and pools for the day and time period selected. Values are shown cumulatively at each IMS checkpoint. Figure 24 shows an example of a report.

This information identifies the report:

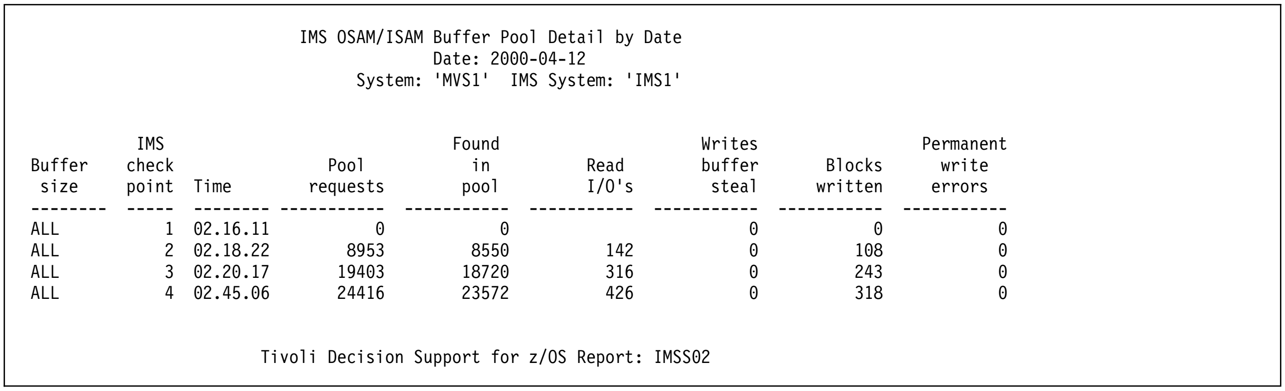
Report ID
IMSS02

Report group
IMS reports

Source
IMS_CHKPT_IOSAM_T

Attributes
IMS, buffers, utilization, performance, daily, OSAM, ISAM

Variables
MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME



```
                             IMS OSAM/ISAM Buffer Pool Detail by Date
                                          Date: 2000-04-12
                                   System: 'MVS1'  IMS System: 'IMS1'


            IMS                          Found                          Writes                  Permanent
Buffer     check                 Pool      in          Read             buffer        Blocks      write
 size      point  Time       requests     pool         I/O's            steal        written     errors
--------   -----  -------- -----------  -----------  -----------  -----------  -----------  -----------
ALL            1  02.16.11           0            0                         0            0            0
ALL            2  02.18.22        8953         8550          142            0          108            0
ALL            3  02.20.17       19403        18720          316            0          243            0
ALL            4  02.45.06       24416        23572          426            0          318            0


                         Tivoli Decision Support for z/OS Report: IMSS02
```

Figure 24. Example of IMS OSAM/ISAM Buffer Pool Detail by Date tabular report

The report contains this information:

Buffer size
The size of the buffers in the buffer pool, or ALL.

IMS check point
The numeric ID of the checkpoint for the IMS system.

Time The time when the activity started, in the form HH.MM.SS.

Pool requests
The number of requests.

Found in pool
The number of requests satisfied from pool (I/O not required).

Read I/O's
The number of read I/O operations performed.

Writes buffer steal
The number of QSAM writes issued (single block writes because of buffer steal).

Blocks written
The number of blocks written by purge requests.

Permanent write errors
The number of permanent write error buffers currently locked in the pool.

IMS VSAM Buffer Pool Detail by Date report

This report shows the utilization of IMS VSAM buffers and pools for the day and time period selected. Values are shown cumulatively at each IMS checkpoint. Figure 25 shows an example of a report.

This information identifies the report:

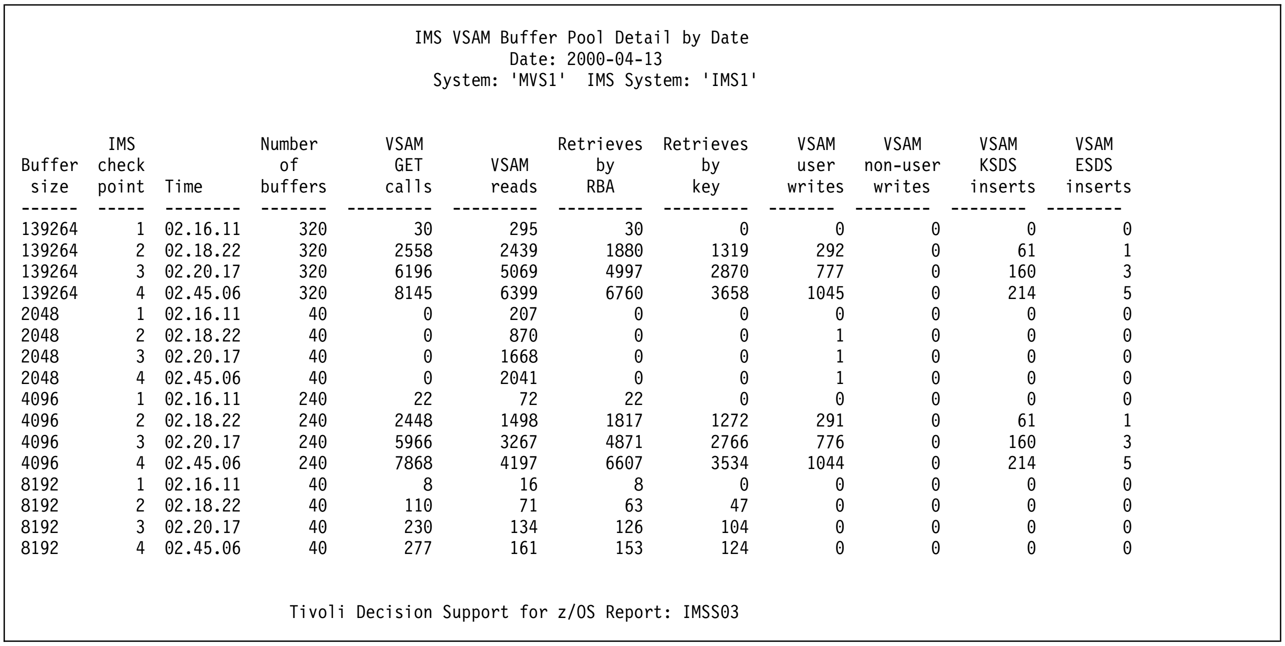
Report ID
IMSS03

Report group
IMS reports

Source
IMS_CHKPT_VSAM_T

Attributes
IMS, buffers, utilization, performance, daily, VSAM

Variables
MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME



```
                                        IMS VSAM Buffer Pool Detail by Date
                                                 Date: 2000-04-13
                                          System: 'MVS1'  IMS System: 'IMS1'


         IMS               Number      VSAM                 Retrieves  Retrieves      VSAM      VSAM      VSAM      VSAM
 Buffer  check               of         GET        VSAM        by         by          user    non-user    KSDS      ESDS
  size   point  Time       buffers     calls       reads      RBA         key        writes    writes    inserts   inserts
 ------  -----  --------   -------   ---------   ---------  ---------  ---------   -------  --------  --------  --------
 139264      1  02.16.11       320          30         295         30          0         0         0         0         0
 139264      2  02.18.22       320        2558        2439       1880       1319       292         0        61         1
 139264      3  02.20.17       320        6196        5069       4997       2870       777         0       160         3
 139264      4  02.45.06       320        8145        6399       6760       3658      1045         0       214         5
 2048        1  02.16.11        40           0         207          0          0         0         0         0         0
 2048        2  02.18.22        40           0         870          0          0         1         0         0         0
 2048        3  02.20.17        40           0        1668          0          0         1         0         0         0
 2048        4  02.45.06        40           0        2041          0          0         1         0         0         0
 4096        1  02.16.11       240          22          72         22          0         0         0         0         0
 4096        2  02.18.22       240        2448        1498       1817       1272       291         0        61         1
 4096        3  02.20.17       240        5966        3267       4871       2766       776         0       160         3
 4096        4  02.45.06       240        7868        4197       6607       3534      1044         0       214         5
 8192        1  02.16.11        40           8          16          8          0         0         0         0         0
 8192        2  02.18.22        40         110          71         63         47         0         0         0         0
 8192        3  02.20.17        40         230         134        126        104         0         0         0         0
 8192        4  02.45.06        40         277         161        153        124         0         0         0         0


                           Tivoli Decision Support for z/OS Report: IMSS03
```

Figure 25. Example of IMS VSAM Buffer Pool Detail by Date tabular report

The report contains this information:

Buffer size
The size of the buffers in the buffer pool.

IMS check point
The numeric ID of the checkpoint for the IMS session.

Time The time of the checkpoint, in the form HH.MM.SS.

Number of buffers
The number of buffers in the buffer pool for the IMS session.

VSAM GET calls
The number of VSAM GET calls issued for the IMS session.

VSAM reads
The number of VSAM read I/O operations for the IMS session.

Retrieves by RBA
The number of requests to retrieve by RBA for the IMS session.

Retrieves by key
The number of requests to retrieve by key for the IMS session.

VSAM user writes
The number of VSAM user write requests for the IMS session.

VSAM non-user writes
The number of VSAM space write requests for the IMS session.

VSAM KSDS inserts
The number of logical records inserted to KSDS for the IMS session.

VSAM ESDS inserts
The number of logical records inserted to ESDS for the IMS session.

IMS Availability reports

IMS availability reports show you the availability of IMS subsystem and region over a specific time period.

IMS CSQ Subsystem Availability, Daily Trend Report

This report shows availability for one IMS subsystem in a daily trend between the FROM_DATE and TO_DATE specified.

This information identifies the report:

Report ID
CSQA01

Report group
CSQ reports

Source
IMS_AVAILABILITY_D

Attributes
IMS, Availability, Daily, Trend

Variables
IMS_System_ID, From_Date, To_Date

```
             IMS CSQ Subsystem Availability, Daily Trend
                Date: '2002-09-27 to '2002-09-30'
                     IMS System: 'IMS 71A

                           Up In       Up In
                    Up     Schedule    Schedule   Objective
     Date        Hours     Hours         %           %
----------  ----------  ----------  ----------  ----------
2002-09-27          12           9      100.00       95.00
2002-09-28          24           9      100.00       95.00
2002-09-29          24           9      100.00       95.00
2002-09-30           3           3     33.33       95.00

     Tivoli Decision Support for z/OS Report: CSQA01
```

Figure 26. Example of an IMS CSQ subsystem Availability, Daily Trend Report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Up Hours
The total time, in hours, when the IMS subsystem was up and running.

Up In Schedule (Hours)
The time within the schedule, in hours, when the IMS subsystem was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Up In Schedule (%)
The time within the schedule, in percent of scheduled hours, when the IMS subsystem was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Objective (%)
Availability objective for this resource in the scheduled hours.

IMS CSQ Region Availability, Daily Overview report

This report gives you a daily overview of the availability of all the IMS regions in an IMS subsystem.

This information identifies the report:

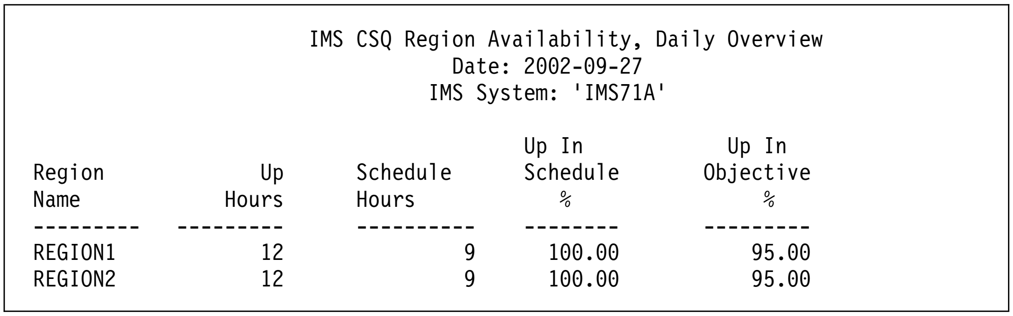
Report ID
CSQA02

Report group
CSQ reports

Source
IMS_AVAILABILITY_D

Attributes
IMS, Availability, Daily, Overview

Variables
IMS_System_ID, Date



```
                IMS CSQ Region Availability, Daily Overview
                              Date: 2002-09-27
                            IMS System: 'IMS71A'

                                                Up In           Up In
Region              Up       Schedule       Schedule        Objective
Name             Hours       Hours             %                %
---------    ---------      ----------     --------        ---------
REGION1             12               9       100.00            95.00
REGION2             12               9       100.00            95.00
```

Figure 27. Example of an IMS CSQ Region Availability, Daily Overview report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Region Name
The name of the IMS region.

Up Hours
The total time, in hours, when the IMS region was up and running.

Up In Schedule (Hours)
The time within the schedule, in hours, when the IMS region was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Up In Schedule (%)
The time within the schedule, in percent of scheduled hours, when the IMS region was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Objective (%)
Availability objective for this resource in the scheduled hours.

IMS CSQ Application Usage and Availability report

This report gives you a daily overview of the availability and CPU usage of all the IMS applications in an IMS subsystem. The availability is based on the IMS subsystem availability.

This information identifies the report:

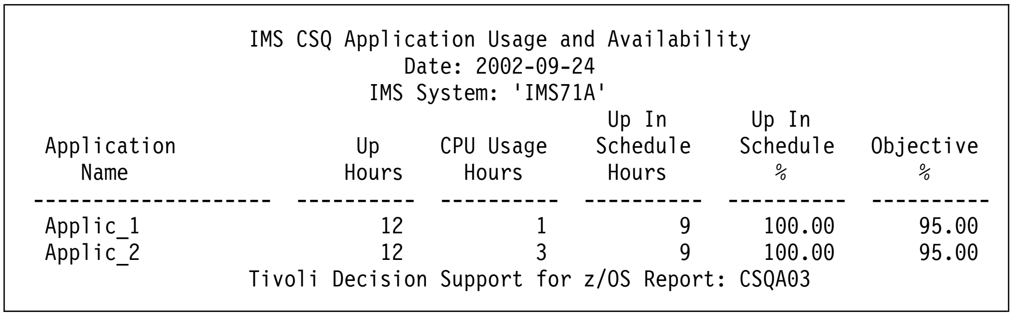
Report ID
CSQA03

Report group
CSQ reports

Source
IMS_AVAILABILITY_D, IMS_PSB_ACCOUNT_D

Attributes
IMS, Availability, Daily, Overview

Variables
IMS_System_ID, Date



```
                    IMS CSQ Application Usage and Availability
                                 Date: 2002-09-24
                              IMS System: 'IMS71A'
                                                   Up In       Up In
 Application             Up      CPU Usage    Schedule    Schedule   Objective
    Name                Hours      Hours        Hours         %           %
--------------------  ----------  ----------  ----------  ----------  ----------
 Applic_1                   12           1           9      100.00       95.00
 Applic_2                   12           3           9      100.00       95.00
                    Tivoli Decision Support for z/OS Report: CSQA03
```

Figure 28. Example of an IMS CSQ application Usage and Availability report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Application Name
The name of the IMS application.

IMS CSQ Resource Utilization, Daily Overview report

This report gives you a daily overview of resource utilization of all the IMS applications in an IMS subsystem.

This information identifies the report:

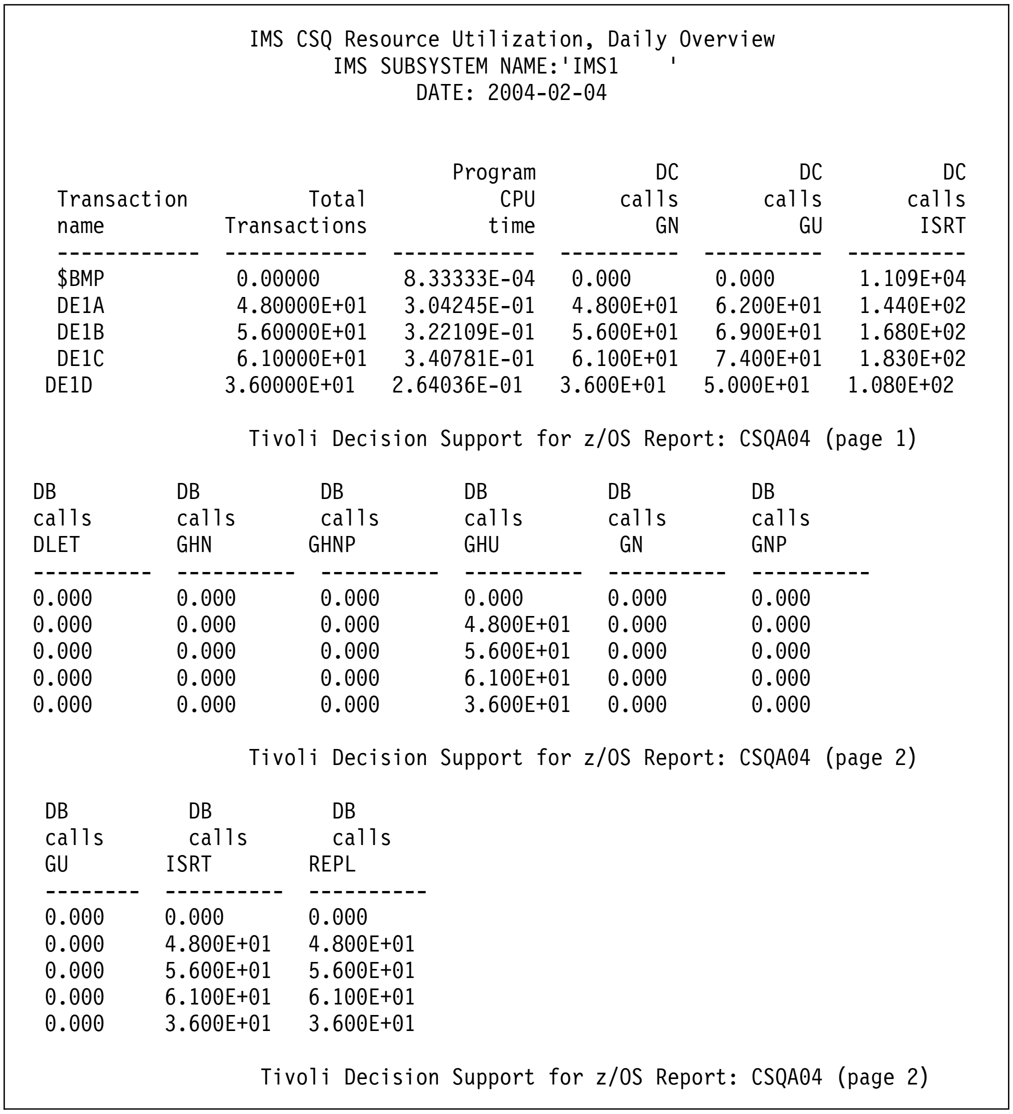
Report ID
CSQA04

Report group
CSQ reports

Source
IMS_PSB_ACCOUNT_D

Attributes
IMS, Resource Utilization, Daily, Overview

Variables
IMS_System_ID, Date



```
                    IMS CSQ Resource Utilization, Daily Overview
                           IMS SUBSYSTEM NAME:'IMS1    '
                                  DATE: 2004-02-04


                                    Program          DC          DC          DC
  Transaction           Total           CPU       calls       calls       calls
  name           Transactions          time          GN          GU        ISRT
  ------------   ------------  ------------  ----------  ----------  ----------
  $BMP             0.00000      8.33333E-04   0.000       0.000       1.109E+04
  DE1A             4.80000E+01  3.04245E-01   4.800E+01   6.200E+01   1.440E+02
  DE1B             5.60000E+01  3.22109E-01   5.600E+01   6.900E+01   1.680E+02
  DE1C             6.10000E+01  3.40781E-01   6.100E+01   7.400E+01   1.830E+02
 DE1D             3.60000E+01  2.64036E-01  3.600E+01   5.000E+01   1.080E+02

                  Tivoli Decision Support for z/OS Report: CSQA04 (page 1)

DB          DB          DB          DB          DB          DB
calls       calls       calls       calls       calls       calls
DLET        GHN        GHNP         GHU          GN         GNP
----------  ----------  ----------  ----------  ----------  ----------
0.000       0.000       0.000       0.000       0.000       0.000
0.000       0.000       0.000       4.800E+01   0.000       0.000
0.000       0.000       0.000       5.600E+01   0.000       0.000
0.000       0.000       0.000       6.100E+01   0.000       0.000
0.000       0.000       0.000       3.600E+01   0.000       0.000

                  Tivoli Decision Support for z/OS Report: CSQA04 (page 2)

 DB          DB          DB
 calls       calls       calls
 GU        ISRT        REPL
 --------  ----------  ----------
 0.000     0.000       0.000
 0.000     4.800E+01   4.800E+01
 0.000     5.600E+01   5.600E+01
 0.000     6.100E+01   6.100E+01
 0.000     3.600E+01   3.600E+01

                   Tivoli Decision Support for z/OS Report: CSQA04 (page 2)
```

Figure 29. Example of an IMS CSQ Resource Utilization, Daily Overview report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Transaction name
The name of the transaction.

Total Transactions
The total number of transactions.

Program CPU time
The total dependent region CPU seconds.

DC calls GN
The total number of DL/I message queue GN calls.

DC calls GU
The total number of DL/I message queue GU calls.

DC calls ISRT
The total number of DL/I message queue ISRT calls.

DB calls DLET
The total number of DL/I database DLET calls.

DB calls GHN
The total number of DL/I database GHN calls.

DB calls GHNP
The total number of DL/I database GHNP calls.

DB calls GHU
The total number of DL/I database GHU calls.

DB calls GN
The total number of DL/I database GN calls.

DB calls GNP
The total number of DL/I database GNP calls.

DB calls GU
The total number of DL/I database GU calls.

DB calls ISRT
The total number of DL/I database ISRT calls.

DB calls REPL
The total number of DL/I database REPL calls.

IMS CSQ Application Usage and Availability report

This report gives you a daily overview of the availability and CPU usage of all the IMS applications in an IMS subsystem. The availability is based on the IMS subsystem availability.

Note: This report is only available if you have installed the Extended Accounting subcomponent.

This information identifies the report:

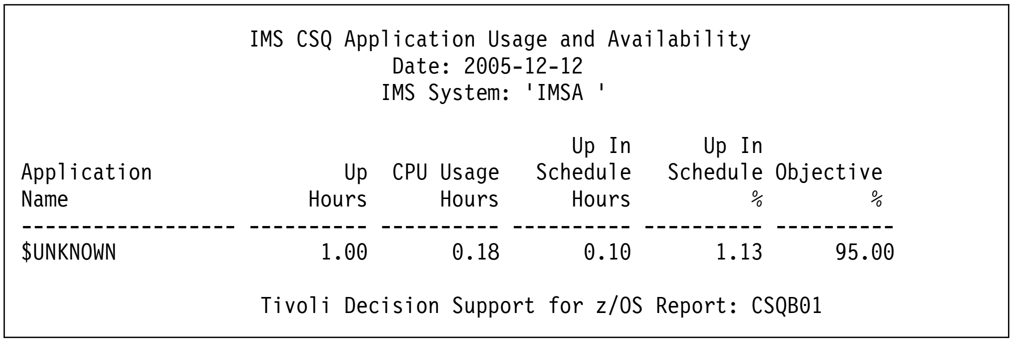
Report ID
CSQB01

Report group
CSQ reports

Source
IMS_AVAILABILITY_D, IMS_PSB_ACCOUNT2_D

Attributes
IMS, Availability, Daily, Overview

Variables
IMS_System_ID, Date



```
                    IMS CSQ Application Usage and Availability
                                Date: 2005-12-12
                               IMS System: 'IMSA '

                                                Up In      Up In
Application                 Up  CPU Usage   Schedule   Schedule Objective
Name                     Hours      Hours      Hours          %          %
------------------ ---------- ---------- ---------- ---------- ----------
$UNKNOWN                  1.00       0.18       0.10       1.13      95.00

                     Tivoli Decision Support for z/OS Report: CSQB01
```

Figure 30. Example of an IMS CSQ application Usage and Availability report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Application name
The name of the IMS application.

Up Hours
The total time, in hours, when the IMS region was up and running.

CPU Usage (Hours)
The processor time for the application, in hours.

Up In Schedule (Hours)
The time within the schedule, in hours, when the IMS subsystem for this application was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Up In Schedule (%)
The time within the schedule, in percent of scheduled hours, when the IMS subsystem for this application was up and running. The IMS_AVAIL_RESOURCE is used to specify the schedule name.

Objective (%)
Availability objective for the IMS subsystem related to this application in the scheduled hours.

IMS CSQ Resource Utilization, Daily Overview report

This report gives you a daily overview of resource utilization of all the IMS applications in an IMS subsystem.

Note: This report is available if you have installed the Extended CSQ component only.

This information identifies the report:

Report ID
CSQB02

Report group
CSQ reports

Source
IMS_PSB_ACCOUNT2_D

Attributes
IMS, Resource Utilization, Daily, Overview

Variables
IMS_System_ID, Date

```
               IMS CSQ Resource Utilization, Daily Overview
                        IMS SUBSYSTEM NAME:'CSSP '
                               DATE: 2004-03-05

                                  Program         DC         DC         DC
Transaction           Total           CPU      calls      calls      calls
name           Transactions          time         GN         GU       ISRT
------------   ------------  ------------ ---------- ---------- ----------
$BMP             0.00000      2.60417E-05  0.000      0.000      0.000
HC9APIA0         4.00000      4.53125E-03  0.000      6.000      4.000
H9FDIS           2.00000      2.96875E-03  0.000      4.000      3.600E+01
IVTNO            2.00000      2.63021E-03  0.000      3.000      2.000
IVTNV            3.00000      3.67187E-03  0.000      4.000      3.000

        DB         DB         DB         DB         DB         DB
     calls      calls      calls      calls      calls      calls
      DLET        GHN       GHNP        GHU         GN        GNP
---------- ---------- ---------- ---------- ---------- ----------
 0.000      0.000      0.000      0.000      0.000      0.000
 0.000      0.000      0.000      0.000      0.000      0.000
 0.000      0.000      0.000      0.000      0.000      0.000
 0.000      0.000      0.000      1.000      0.000      0.000
 0.000      0.000      0.000      1.000      0.000      0.000

   DB         DB         DB
calls      calls      calls
   GU       ISRT       REPL
----- ---------- ----------
0.000    0.000      0.000
0.000    0.000      0.000
0.000    0.000      0.000
1.000    0.000      1.000
2.000    0.000      1.000

                         Tivoli Decision Support Report: CSQB02
```

Figure 31. Example of an IMS CSQ Resource Utilization, Daily Overview report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem.

Transaction Name
The name of the transaction.

Total Transactions
The total number of transactions.

Program CPU time
The total dependent region CPU seconds.

DC calls GN
The total number of DL/I message queue GN calls.

DC calls GU
The total number of DL/I message queue GU calls.

DC calls ISRT
The total number of DL/I message queue ISRT calls.

DB calls DLET
The total number of DL/I database DLET calls.

DB calls GHN
The total number of DL/I database GHN calls.

DB calls GHNP
The total number of DL/I database GHNP calls.

DB calls GHU
The name of the transaction.

DB calls GN
The total number of DL/I database GN calls.

DB calls GNP
The name of the transaction.

DB calls GU
The total number of DL/I database GU calls.

DB calls ISRT
The name of the transaction.

DB calls REPL
The total number of DL/I database REPL calls.

IMS Message Queue reports

These reports show statistics of the IMS Message Queue utilization.

IMS Message Queue Utilization, Date report

This report shows the Input and Output message queue utilization for the selected IMS system and Queue type, at 15-minute intervals during a specified time period. Figure 32 on page 129 shows an example of a report.

This information identifies the report:

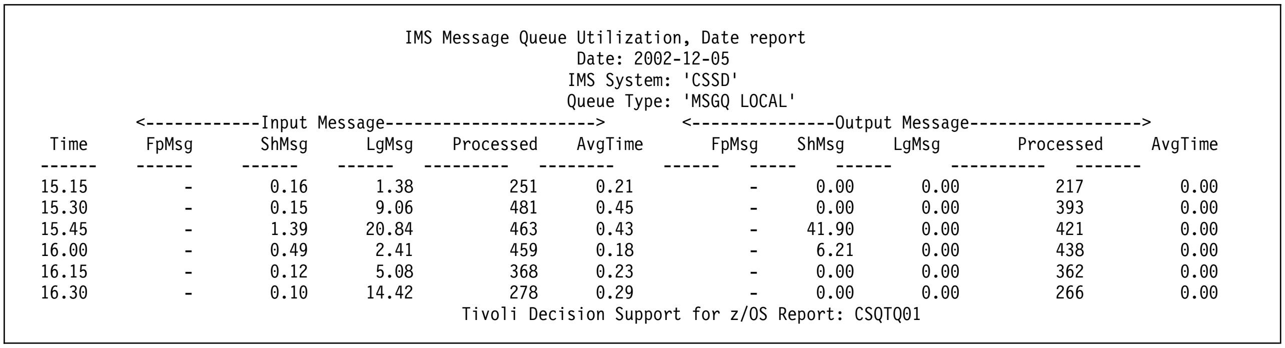
Report ID
CSQTQ01

Report group
CSQ reports

Source
IMS_TRAN_QUEUE_QV

Attributes
IMS, Transaction, Queue, Date

Variables
Origin_IMS, Date, Queue_Type



```
                                      IMS Message Queue Utilization, Date report
                                                        Date: 2002-12-05
                                                       IMS System: 'CSSD'
                                                       Queue Type: 'MSGQ LOCAL'
          <------------Input Message----------------------->       <---------------Output Message------------------->
 Time      FpMsg       ShMsg      LgMsg    Processed    AvgTime       FpMsg    ShMsg     LgMsg        Processed     AvgTime
------    ------     ------    ------   ---------   --------     ------   -----   ------       ----------   -------
15.15          -         0.16       1.38          251      0.21           -       0.00       0.00          217          0.00
15.30          -         0.15       9.06          481      0.45           -       0.00       0.00          393          0.00
15.45          -         1.39      20.84          463      0.43           -      41.90       0.00          421          0.00
16.00          -         0.49       2.41          459      0.18           -       6.21       0.00          438          0.00
16.15          -         0.12       5.08          368      0.23           -       0.00       0.00          362          0.00
16.30          -         0.10      14.42          278      0.29           -       0.00       0.00          266          0.00
                                            Tivoli Decision Support for z/OS Report: CSQTQ01
```

Figure 32. Example of IMS Message Queue Utilization, Date report

The report contains this information:

Date The date of the day for the measurement.

Time The 15-minute interval for which the data applies.

IMS System
The name of the IMS subsystem.

Queue Type
The queue type.

Input Message
Statistics on inbound message queues activities.

FpMsg
Average Fast Path messages on queue before processing.

ShMsg
Average Full Function Short messages on queue before processing.

LgMsg
Average Full Function Long messages on queue before processing.

Processed
Total number of messages processed.

AvgTime
Average time (in seconds) each message remains on queue.

Output Message
Statistics on outbound message queues activities.

FpMsg
Average Fast Path messages on queue before processing.

ShMsg
Average Full Function Short messages on queue before processing.

LgMsg
Average Full Function Long messages on queue before processing.

Processed
Total number of messages processed.

AvgTime
Average time (in seconds) each message remains on queue.

IMS Msg Queue Utilization by Transaction, Date report

This report shows the Input and Output message queue utilization by transaction name for the selected IMS system and Queue type, at 15-minute intervals during a specified time period. Figure 33 shows an example of a report.

This information identifies the report:

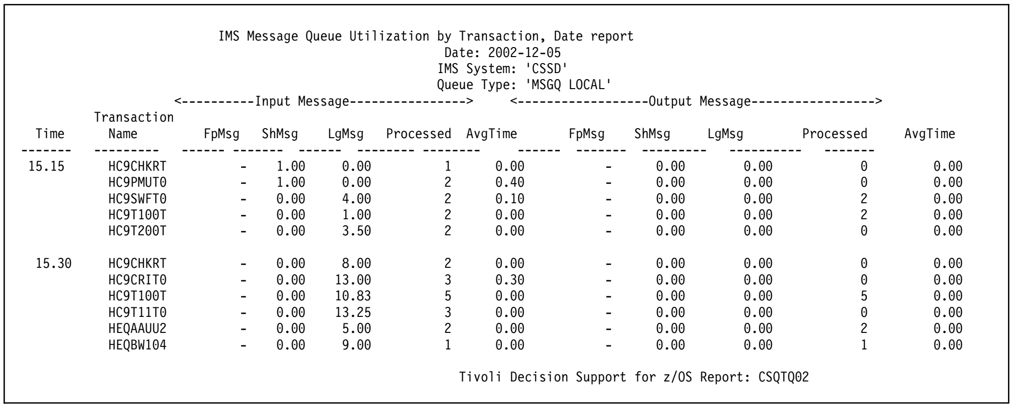
Report ID
CSQTQ02

Report group
CSQ reports

Source
IMS_TRAN_QUEUE_QV

Attributes
IMS, Transaction, Queue, Date

Variables
Origin_IMS, Date, Queue_Type



```
                           IMS Message Queue Utilization by Transaction, Date report
                                                          Date: 2002-12-05
                                                         IMS System: 'CSSD'
                                                         Queue Type: 'MSGQ LOCAL'
                     <----------Input Message---------------->     <------------------Output Message----------------->
          Transaction
  Time      Name         FpMsg   ShMsg    LgMsg   Processed  AvgTime       FpMsg    ShMsg      LgMsg        Processed     AvgTime
-------   ---------   ------ -------  ------  -------- --------     ------  -------  ---------    ----------   -------
 15.15      HC9CHKRT          -    1.00     0.00          1      0.00            -     0.00         0.00            0         0.00
            HC9PMUT0          -    1.00     0.00          2      0.40            -     0.00         0.00            0         0.00
            HC9SWFT0          -    0.00     4.00          2      0.10            -     0.00         0.00            2         0.00
            HC9T100T          -    0.00     1.00          2      0.00            -     0.00         0.00            2         0.00
            HC9T200T          -    0.00     3.50          2      0.00            -     0.00         0.00            0         0.00

  15.30     HC9CHKRT          -    0.00     8.00          2      0.00            -     0.00         0.00            0         0.00
            HC9CRIT0          -    0.00    13.00          3      0.30            -     0.00         0.00            0         0.00
            HC9T100T          -    0.00    10.83          5      0.00            -     0.00         0.00            5         0.00
            HC9T11T0          -    0.00    13.25          3      0.00            -     0.00         0.00            0         0.00
            HEQAAUU2          -    0.00     5.00          2      0.00            -     0.00         0.00            2         0.00
            HEQBW104          -    0.00     9.00          1      0.00            -     0.00         0.00            1         0.00

                                                            Tivoli Decision Support for z/OS Report: CSQTQ02
```

Figure 33. Example of IMS Message Queue Utilization by Transaction, Date report

The report contains this information:

Date The date of the day for the measurement.

Time The 15-minute interval for which the data applies.

IMS System
The name of the IMS subsystem.

Queue Type
The queue type.

Transaction Name
The name of the IMS transaction.

For the description of the other columns in this report, see "IMS Message Queue Utilization, Date report" on page 128.

IMS Message Queue Utilization Overview, Daily Report

This report shows an overview of the different queue types utilization for the selected IMS system between the FROM_DATE and TO_DATE specified. Figure 34 on page 131 shows an example of a report.

This information identifies the report:

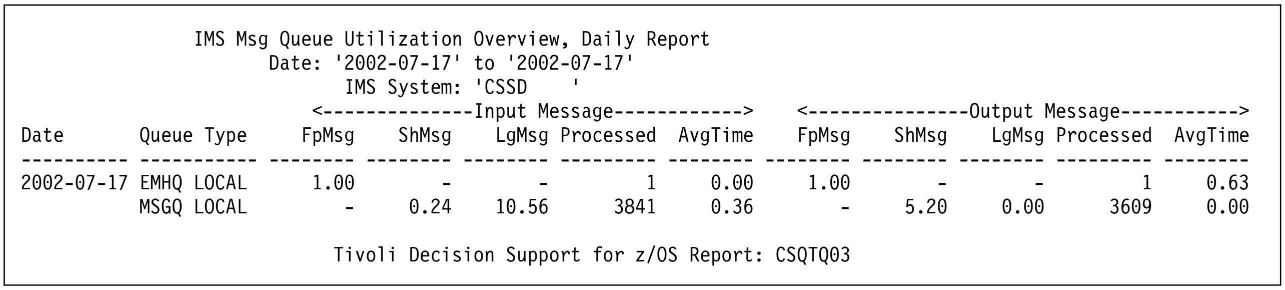
Report ID
CSQTQ03

Report group
CSQ reports

Source
IMS_TRAN_QUEUE_DV

Attributes
IMS, Transaction, Queue, Date

Variables
Origin_IMS, From_Date, To_Date



```
                IMS Msg Queue Utilization Overview, Daily Report
                       Date: '2002-07-17' to '2002-07-17'
                              IMS System: 'CSSD    '
                           <--------------Input Message------------>    <---------------Output Message----------->
Date       Queue Type     FpMsg    ShMsg    LgMsg Processed  AvgTime    FpMsg    ShMsg    LgMsg Processed  AvgTime
---------- ----------- -------- -------- -------- --------- -------- -------- -------- -------- --------- --------
2002-07-17 EMHQ LOCAL      1.00        -        -         1     0.00     1.00        -        -         1     0.63
           MSGQ LOCAL         -     0.24    10.56      3841     0.36        -     5.20     0.00      3609     0.00

                             Tivoli Decision Support for z/OS Report: CSQTQ03
```

Figure 34. Example of an IMS Message Queue Utilization Overview, Daily Report

The report contains this information:

Date The date of the day for the measurement.

IMS System
The name of the IMS subsystem.

Queue Type
The queue type.

For the description of the other columns in this report, see “IMS Message Queue Utilization, Date report” on page 128.

IMS Transaction Arrival Rate and Msg Queue Usage, Daily Trend

This report shows a daily trend on how the selected IMS system performs with the message queue resources of the selected queue type, compared with the transaction arrival rate, between the FROM_DATE and TO_DATE specified. Figure 35 on page 132 shows an example of a report.

This information identifies the report:

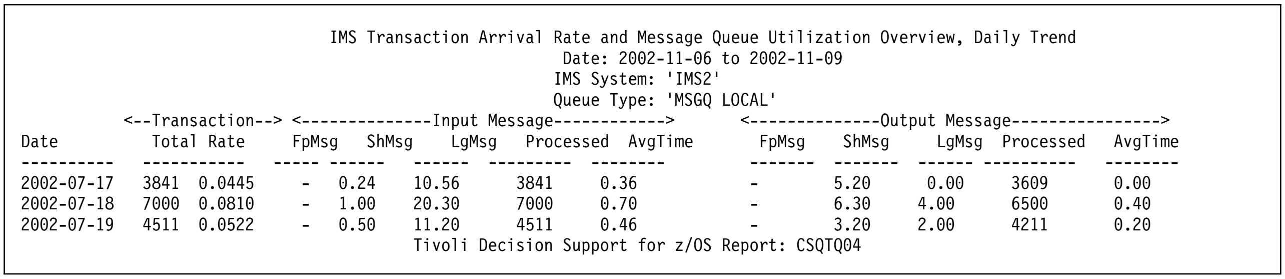
Report ID
CSQTQ04

Report group
CSQ reports

Source
IMS_TRAN_QUEUE_DV

Attributes
IMS, Transaction, Queue, Date

Variables
Origin_IMS, From_Date, To_Date, Queue Type



```
                                 IMS Transaction Arrival Rate and Message Queue Utilization Overview, Daily Trend
                                                          Date: 2002-11-06 to 2002-11-09
                                                         IMS System: 'IMS2'
                                                         Queue Type: 'MSGQ LOCAL'
           <--Transaction--> <--------------Input Message------------>      <--------------Output Message---------------->
Date          Total Rate     FpMsg   ShMsg    LgMsg   Processed  AvgTime       FpMsg    ShMsg     LgMsg  Processed   AvgTime
----------   -----------   ----- ------   ------  --------- --------         -------   -------  ------ ----------    --------
2002-07-17   3841  0.0445     -   0.24    10.56      3841     0.36            -         5.20     0.00     3609       0.00
2002-07-18   7000  0.0810     -   1.00    20.30      7000     0.70            -         6.30    4.00      6500       0.40
2002-07-19   4511  0.0522     -   0.50    11.20      4511     0.46            -         3.20    2.00      4211       0.20
                                          Tivoli Decision Support for z/OS Report: CSQTQ04
```

Figure 35. Example of an IMS Transaction Arrival Rate and Message Queue Usage, Daily Trend

The report contains this information:

Date The date of the day for the measurement.

IMS System
The name of the IMS subsystem.

Queue Type
The queue type.

Transaction Total
The total number of transactions processed.

Transaction Rate
The total arrival date (transactions per second).

For the description of the other columns, see “IMS Message Queue Utilization, Date report” on page 128.

IMS CSQ Transaction Transit Time reports

These reports show statistics of the IMS Transaction Transit Time performance.

IMS CSQ Transit Time Analysis By Transaction Name

This daily report shows the transaction transit time metrics for the selected IMS systems, date, and Transaction Name. Figure 36 on page 133 shows an example of a report.

This information identifies the report:

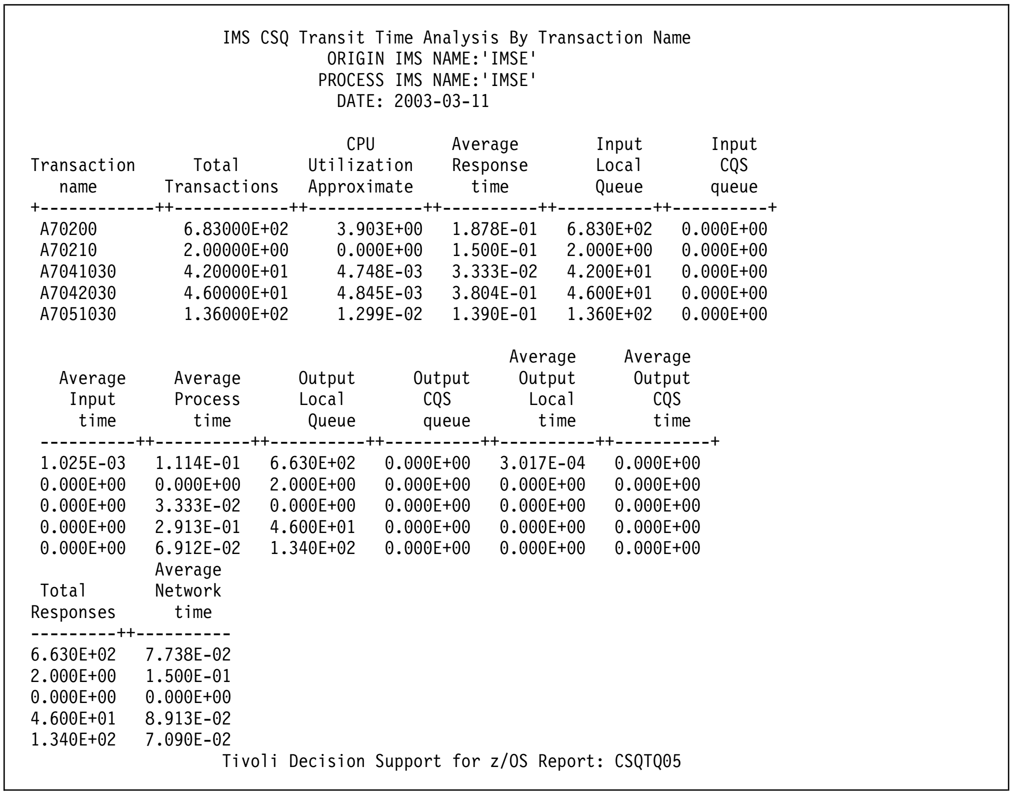
Report ID
CSQTQ05

Report group
CSQ reports

Source
IMS_TRAN_D

Attributes
IMS, Transaction, Utilization, Performance, Daily

Variables
Origin_IMS, Process_IMS, Date, Transaction_Name



```
                    IMS CSQ Transit Time Analysis By Transaction Name
                               ORIGIN IMS NAME:'IMSE'
                              PROCESS IMS NAME:'IMSE'
                                DATE: 2003-03-11

                                 CPU        Average        Input       Input
Transaction      Total       Utilization    Response       Local        CQS
   name       Transactions   Approximate      time         Queue       queue
+------------++------------++------------++----------++----------++----------+
 A70200         6.83000E+02     3.903E+00   1.878E-01   6.830E+02   0.000E+00
 A70210         2.00000E+00     0.000E+00   1.500E-01   2.000E+00   0.000E+00
 A7041030       4.20000E+01     4.748E-03   3.333E-02   4.200E+01   0.000E+00
 A7042030       4.60000E+01     4.845E-03   3.804E-01   4.600E+01   0.000E+00
 A7051030       1.36000E+02     1.299E-02   1.390E-01   1.360E+02   0.000E+00

                                                   Average     Average
   Average     Average      Output      Output      Output      Output
    Input      Process      Local        CQS        Local        CQS
     time        time        Queue       queue        time        time
 ----------++----------++----------++----------++----------++----------+
 1.025E-03   1.114E-01   6.630E+02   0.000E+00   3.017E-04   0.000E+00
 0.000E+00   0.000E+00   2.000E+00   0.000E+00   0.000E+00   0.000E+00
 0.000E+00   3.333E-02   0.000E+00   0.000E+00   0.000E+00   0.000E+00
 0.000E+00   2.913E-01   4.600E+01   0.000E+00   0.000E+00   0.000E+00
 0.000E+00   6.912E-02   1.340E+02   0.000E+00   0.000E+00   0.000E+00
             Average
 Total       Network
Responses      time
---------++----------
6.630E+02   7.738E-02
2.000E+00   1.500E-01
0.000E+00   0.000E+00
4.600E+01   8.913E-02
1.340E+02   7.090E-02
                    Tivoli Decision Support for z/OS Report: CSQTQ05
```

Figure 36. Example of IMS CSQ Transit Time Analysis By Transaction Name, Daily report

The report contains this information:

ORIGIN IMS
The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin. In a non-CSQ configuration it always matches the PROCESS_IMS value.

PROCESS IMS
The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor. In a non-Shared Queue configuration it always matches the ORIGIN_IMS value.

Date The date of the day for the measurement.

Transaction Name
The name of the IMS transaction the user requested.

Total Transactions
Total number of transactions.

CPU Utilization Approximate
This column represents the sum of approximate number of CPU seconds of program execution time while the transactions were active. This value is not provided for WFI or PWFI transactions (to get the correct value look at the PSB_ACCOUNT_x table or CSQA04 report_ID).

Average Response time
The average time in seconds, needed to process a transaction from the beginning to the end. It should be considered as the sum of the host time plus network time.

Input Local Queue
The total number of input messages issued by transactions, BMP programs, not using CSQ.

Input CQS queue
The total number of input messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non-CSQ configuration.

Average Input time
The average time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions. In a CSQ configuration it also includes the time the transaction spent in SQ before being processed.

Average Process time
The average elapsed time in seconds that transactions and BMP programs spent processing in the dependent regions, in seconds.

Output Local Queue
The total number of output messages issued by transactions and BMP programs, not using CSQ.

Output CQS Queue
The total number of output messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non-CSQ configuration.

Average Output Local time
The average time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.

Average Output CQS time
The time between when the completed output transaction was put on the queue and when it was routed as output to the terminal. Always zero in a non-CSQ configuration.

Total Responses
The total number of SNA definite responses or exception responses for which the message is dequeued.

Average Network time
The average time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction defined as such in IMS.

IMS CSQ Transit Time Analysis By LTERM and Userid

This daily report shows the transaction transit time metrics for the selected IMS systems, date, origin LTERM, and Userid. The summary average transit time values (in bold) will be provided only if you are using Tivoli Decision Support for z/OS with QMF (QMFUSE=YES coded in the Tivoli Decision Support for z/OS initialization member DRLFPROF). Figure 37 on page 136 shows an example of a report.

This information identifies the report:

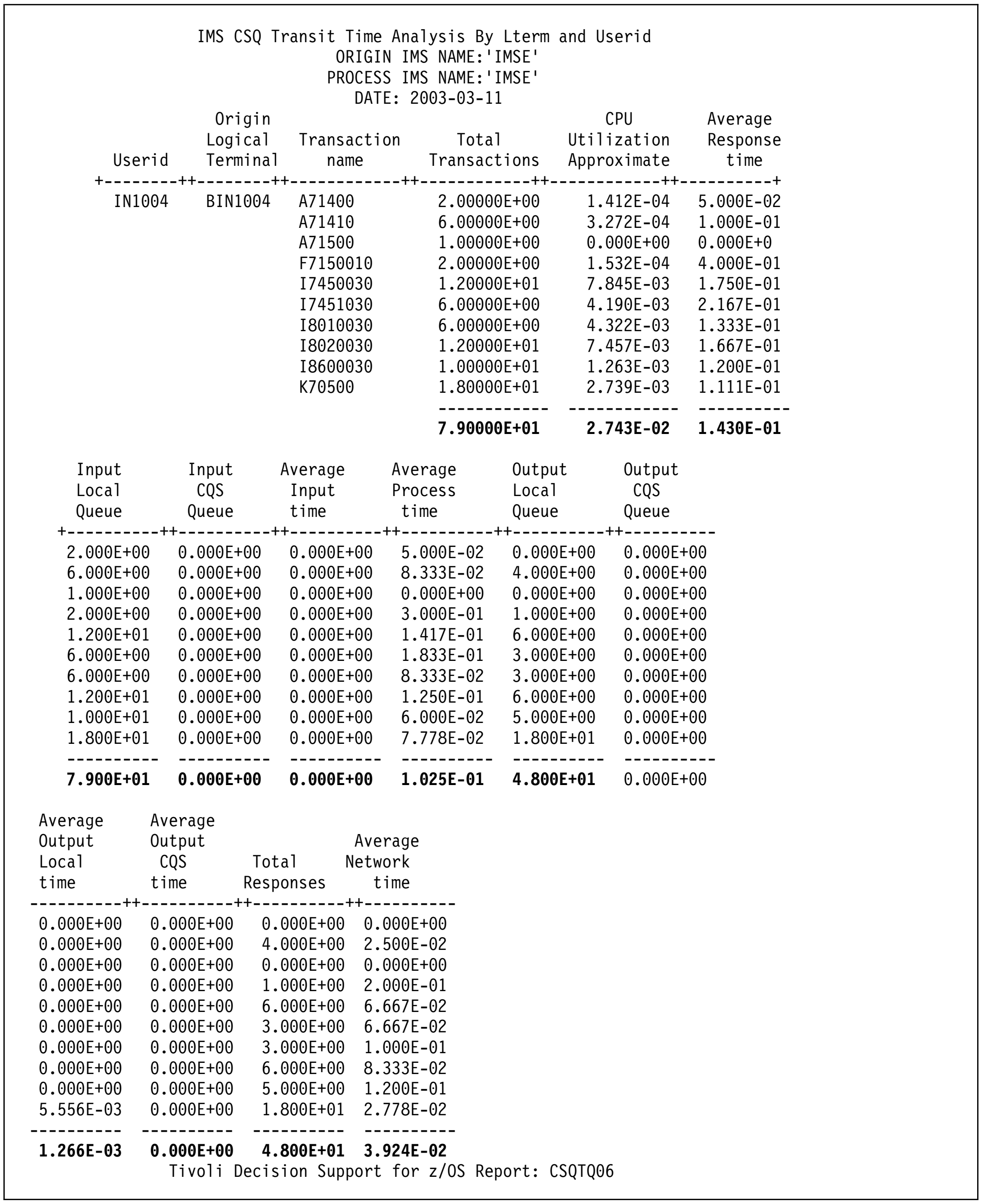
Report ID
CSQTQ06

Report group
CSQ reports

Source
IMS_TRAN_D

Attributes
IMS, Lterm, Utilization, Performance, Daily, User_ID

Variables
Origin_IMS, Process_IMS, Date, origin_Lterm, User_ID



```
                 IMS CSQ Transit Time Analysis By Lterm and Userid
                                ORIGIN IMS NAME:'IMSE'
                               PROCESS IMS NAME:'IMSE'
                                  DATE: 2003-03-11
                  Origin                                    CPU        Average
                  Logical   Transaction     Total       Utilization    Response
         Userid   Terminal     name       Transactions   Approximate      time
       +--------++--------++------------++------------++------------++----------+
         IN1004   BIN1004   A71400        2.00000E+00     1.412E-04   5.000E-02
                            A71410        6.00000E+00     3.272E-04   1.000E-01
                            A71500        1.00000E+00     0.000E+00   0.000E+0
                            F7150010      2.00000E+00     1.532E-04   4.000E-01
                            I7450030      1.20000E+01     7.845E-03   1.750E-01
                            I7451030      6.00000E+00     4.190E-03   2.167E-01
                            I8010030      6.00000E+00     4.322E-03   1.333E-01
                            I8020030      1.20000E+01     7.457E-03   1.667E-01
                            I8600030      1.00000E+01     1.263E-03   1.200E-01
                            K70500        1.80000E+01     2.739E-03   1.111E-01
                                          ------------  ------------  ----------
                                          7.90000E+01     2.743E-02   1.430E-01

     Input       Input     Average     Average      Output      Output
     Local        CQS       Input      Process      Local        CQS
     Queue       Queue      time        time        Queue       Queue
   +----------++----------++----------++----------++----------++----------
    2.000E+00   0.000E+00   0.000E+00   5.000E-02   0.000E+00   0.000E+00
    6.000E+00   0.000E+00   0.000E+00   8.333E-02   4.000E+00   0.000E+00
    1.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
    2.000E+00   0.000E+00   0.000E+00   3.000E-01   1.000E+00   0.000E+00
    1.200E+01   0.000E+00   0.000E+00   1.417E-01   6.000E+00   0.000E+00
    6.000E+00   0.000E+00   0.000E+00   1.833E-01   3.000E+00   0.000E+00
    6.000E+00   0.000E+00   0.000E+00   8.333E-02   3.000E+00   0.000E+00
    1.200E+01   0.000E+00   0.000E+00   1.250E-01   6.000E+00   0.000E+00
    1.000E+01   0.000E+00   0.000E+00   6.000E-02   5.000E+00   0.000E+00
    1.800E+01   0.000E+00   0.000E+00   7.778E-02   1.800E+01   0.000E+00
    ----------  ----------  ----------  ----------  ----------  ----------
    7.900E+01   0.000E+00   0.000E+00   1.025E-01   4.800E+01   0.000E+00

 Average     Average
 Output      Output                 Average
 Local        CQS       Total      Network
 time        time      Responses     time
----------++----------++----------++----------
 0.000E+00   0.000E+00   0.000E+00  0.000E+00
 0.000E+00   0.000E+00   4.000E+00  2.500E-02
 0.000E+00   0.000E+00   0.000E+00  0.000E+00
 0.000E+00   0.000E+00   1.000E+00  2.000E-01
 0.000E+00   0.000E+00   6.000E+00  6.667E-02
 0.000E+00   0.000E+00   3.000E+00  6.667E-02
 0.000E+00   0.000E+00   3.000E+00  1.000E-01
 0.000E+00   0.000E+00   6.000E+00  8.333E-02
 0.000E+00   0.000E+00   5.000E+00  1.200E-01
 5.556E-03   0.000E+00   1.800E+01  2.778E-02
----------  ----------  ----------  ----------
 1.266E-03   0.000E+00   4.800E+01  3.924E-02
              Tivoli Decision Support for z/OS Report: CSQTQ06
```

Figure 37. Example of IMS CSQ Transit Time Analysis By LTERM and Userid, Daily report

The report contains this information:

ORIGIN IMS
The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin. In a non-CSQ configuration it always matches the PROCESS_IMS value.

PROCESS IMS
The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor. In a non-Shared Queue configuration it always matches the ORIGIN_IMS value.

Date The date of the day for the measurement.

Userid
The user identifier used to gain authorized access to IMS resources. This column contains the logical terminal name if security is not being managed by the IMS-supported /SIGNON.

Origin Logical Terminal
The IMS-defined logical name for the terminal used to request the transaction or OTMA Tpipe name.

Transaction Name
The name of the IMS transaction the user requested.

Total Transactions
Total number of transactions.

CPU Utilization Approximate
This column represents the sum of approximate number of CPU seconds of program execution time while the transactions were active. This value is not provided for WFI or PWFI transactions (to get the correct value look at the PSB_ACCOUNT_x table or CSQA04 report ID).

Average Response time
The average time in seconds, needed to process a transaction from the beginning to the end. It should be considered as the sum of the host time plus network time.

Input Local Queue
The total number of input messages issued by transactions, BMP programs, not using CSQ.

Input CQS queue
The total number of input messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non-CSQ configuration.

Average Input time
The average time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions. In a CSQ configuration it also includes the time the transaction spent in SQ before being processed.

Average Process time
The average elapsed time in seconds that transactions and BMP programs spent processing in the dependent regions, in seconds.

Output Local Queue
The total number of output messages issued by transactions and BMP programs, not using CSQ.

Output CQS Queue
The total number of output messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non- CSQ configuration.

Average Output Local time
The average time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.

Average Output CQS time
The time between when the completed output transaction was put on the queue and when it was routed the output to the terminal. Always zero in a non CSQ configuration.

Total Responses
The total number of SNA definite responses or exception responses for which the message is dequeued.

Average Network time
The average time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction is defined as such in IMS.

IMS CSQ Transit Time Analysis by Region

This daily report shows the transaction transit time metrics for the selected IMS systems, date, and Region Job Name. The summary average transit time values (in bold) will be provided only if you are using Tivoli Decision Support for z/OS with QMF (QMFUSE=YES coded in the Tivoli Decision Support for z/OS initialization member DRLFPROF). Figure 38 on page 139 shows an example of a report.

This information identifies the report:

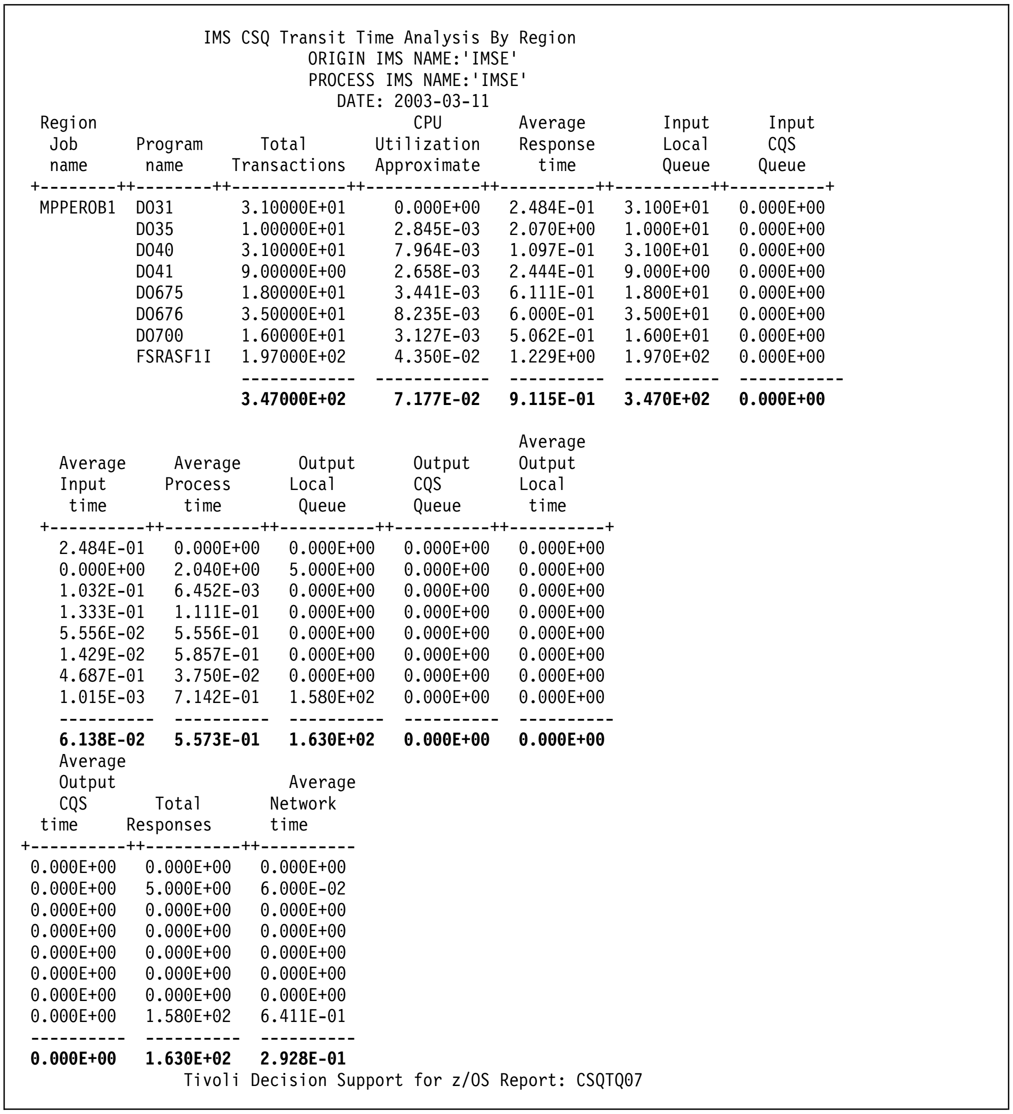
Report ID
CSQTQ07

Report group
CSQ reports

Source
IMS_TRAN_D

Attributes
IMS, Region, Utilization, Performance, Daily

Variables
Origin_IMS, Process_IMS, Date, Region_Job_Name



```
                    IMS CSQ Transit Time Analysis By Region
                               ORIGIN IMS NAME:'IMSE'
                               PROCESS IMS NAME:'IMSE'
                                  DATE: 2003-03-11
  Region                                  CPU        Average        Input      Input
   Job      Program      Total        Utilization    Response       Local       CQS
   name      name     Transactions    Approximate      time         Queue      Queue
 +--------++--------++------------++------------++----------++----------++----------+
  MPPEROB1  DO31       3.10000E+01     0.000E+00   2.484E-01   3.100E+01   0.000E+00
            DO35       1.00000E+01     2.845E-03   2.070E+00   1.000E+01   0.000E+00
            DO40       3.10000E+01     7.964E-03   1.097E-01   3.100E+01   0.000E+00
            DO41       9.00000E+00     2.658E-03   2.444E-01   9.000E+00   0.000E+00
            DO675      1.80000E+01     3.441E-03   6.111E-01   1.800E+01   0.000E+00
            DO676      3.50000E+01     8.235E-03   6.000E-01   3.500E+01   0.000E+00
            DO700      1.60000E+01     3.127E-03   5.062E-01   1.600E+01   0.000E+00
            FSRASF1I   1.97000E+02     4.350E-02   1.229E+00   1.970E+02   0.000E+00
                       ------------  ------------  ----------  ----------  -----------
                       3.47000E+02     7.177E-02   9.115E-01   3.470E+02   0.000E+00

                                                    Average
    Average     Average      Output      Output     Output
    Input      Process      Local        CQS        Local
     time        time        Queue       Queue       time
  +----------++----------++----------++----------++----------+
    2.484E-01   0.000E+00   0.000E+00   0.000E+00   0.000E+00
    0.000E+00   2.040E+00   5.000E+00   0.000E+00   0.000E+00
    1.032E-01   6.452E-03   0.000E+00   0.000E+00   0.000E+00
    1.333E-01   1.111E-01   0.000E+00   0.000E+00   0.000E+00
    5.556E-02   5.556E-01   0.000E+00   0.000E+00   0.000E+00
    1.429E-02   5.857E-01   0.000E+00   0.000E+00   0.000E+00
    4.687E-01   3.750E-02   0.000E+00   0.000E+00   0.000E+00
    1.015E-03   7.142E-01   1.580E+02   0.000E+00   0.000E+00
    ----------  ----------  ----------  ----------  ----------
    6.138E-02   5.573E-01   1.630E+02   0.000E+00   0.000E+00
    Average
    Output                  Average
    CQS       Total        Network
  time     Responses      time
 +----------++----------++----------
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   5.000E+00   6.000E-02
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00
  0.000E+00   1.580E+02   6.411E-01
  ----------  ----------  ----------
  0.000E+00   1.630E+02   2.928E-01
                  Tivoli Decision Support for z/OS Report: CSQTQ07
```

Figure 38. Example of IMS CSQ Transit Time Analysis By Region, Daily report

The report contains this information:

ORIGIN IMS
The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin. In a non-CSQ configuration it always matches the PROCESS_IMS value.

PROCESS IMS
The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor. In a non-Shared Queue configuration it always matches the ORIGIN_IMS value.

Date The date of the day for the measurement.

Region Job Name
The MVS- and JES-identified job name for the IMS dependent region.

Program Name
The name of the IMS application program used to process the transaction.

Total Transactions
Total number of transactions.

CPU Utilization Approximate
This column represents the sum of approximate number of CPU seconds of program execution time while the transactions were active. This value is not provided for WFI or PWFI transactions (to get the correct value look at the PSB_ACCOUNT_x table or CSQA04 report ID).

Average Response time
The average time in seconds, needed to process a transaction from the beginning to the end. It should be considered as the sum of the host time plus network time.

Input Local Queue
The total number of input messages issued by transactions, BMP programs, not using CSQ.

Input CQS queue
The total number of input messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non-CSQ configuration.

Average Input time
The average time, in seconds, that transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions. In a CSQ configuration it also includes the time transaction spent in SQ before being processed.

Average Process time
The average elapsed time in seconds that transactions and BMP programs spent processing in the dependent regions, in seconds.

Output Local Queue
The total number of output messages issued by transactions and BMP programs, not using CSQ.

Output CQS Queue
The total number of output messages issued by transactions and BMP programs queued through IMS CSQ. Always zero in a non-CSQ configuration.

Average Output Local time
The average time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.

Average Output CQS time
The time between when the completed output transaction was put on the queue and when it was routed as the output to the terminal. Always zero in a non-CSQ configuration.

Total Responses
The total number of SNA definite responses or exception responses for which the message is dequeued.

Average Network time
The average time that responding transactions spent in network transmission to the ultimate destination, in seconds, as measured using SNA definite response. This may also include user think time to the next transaction, if the transaction is defined as such in IMS.

IMS CSQ Utilization reports

IMS utilization reports show you the metrics of system resource utilization.

IMS CSQ Resource Utilization, Daily Overview

IMS CSQ utilization reports show you the metrics of system resource utilization for the selected IMS ID and date by transaction code. Figure 40 on page 144 shows an example of a report.

This information identifies the report:

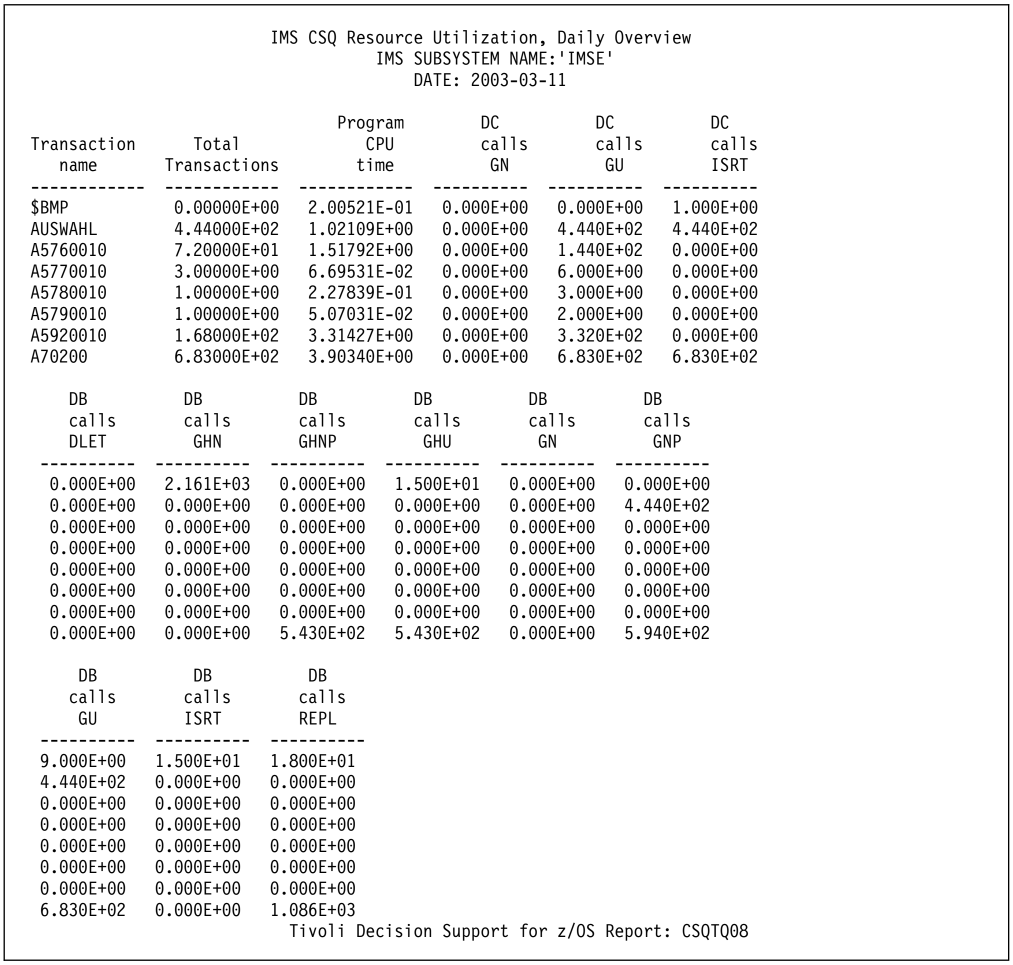
Report ID
CSQTQ08

Report group
CSQ reports

Source
IMS_PSB_ACCOUNT_D

Attributes
IMS, Accounting, Utilization, Daily, Overview

Variables
IMS_System_ID, Date



```
                          IMS CSQ Resource Utilization, Daily Overview
                                 IMS SUBSYSTEM NAME:'IMSE'
                                     DATE: 2003-03-11

                                  Program       DC          DC          DC
Transaction      Total              CPU         calls       calls       calls
   name       Transactions         time          GN          GU         ISRT
------------  ------------  ------------  ----------  ----------  ----------
$BMP           0.00000E+00   2.00521E-01   0.000E+00   0.000E+00   1.000E+00
AUSWAHL        4.44000E+02   1.02109E+00   0.000E+00   4.440E+02   4.440E+02
A5760010       7.20000E+01   1.51792E+00   0.000E+00   1.440E+02   0.000E+00
A5770010       3.00000E+00   6.69531E-02   0.000E+00   6.000E+00   0.000E+00
A5780010       1.00000E+00   2.27839E-01   0.000E+00   3.000E+00   0.000E+00
A5790010       1.00000E+00   5.07031E-02   0.000E+00   2.000E+00   0.000E+00
A5920010       1.68000E+02   3.31427E+00   0.000E+00   3.320E+02   0.000E+00
A70200         6.83000E+02   3.90340E+00   0.000E+00   6.830E+02   6.830E+02

    DB          DB          DB          DB          DB          DB
    calls       calls       calls       calls       calls       calls
    DLET        GHN         GHNP         GHU         GN          GNP
 ----------  ----------  ----------  ----------  ----------  ----------
  0.000E+00   2.161E+03   0.000E+00   1.500E+01   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   4.440E+02
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00   0.000E+00
  0.000E+00   0.000E+00   5.430E+02   5.430E+02   0.000E+00   5.940E+02

     DB          DB          DB
    calls       calls       calls
     GU          ISRT        REPL
 ----------  ----------  ----------
 9.000E+00   1.500E+01   1.800E+01
 4.440E+02   0.000E+00   0.000E+00
 0.000E+00   0.000E+00   0.000E+00
 0.000E+00   0.000E+00   0.000E+00
 0.000E+00   0.000E+00   0.000E+00
 0.000E+00   0.000E+00   0.000E+00
 0.000E+00   0.000E+00   0.000E+00
 6.830E+02   0.000E+00   1.086E+03
                              Tivoli Decision Support for z/OS Report: CSQTQ08
```

Figure 39. Example of IMS CSQ Resource Utilization, Daily Overview report

The report contains this information:

Date The date of the day for the measurement.

IMS System ID
The name of the IMS subsystem effectively processing the transaction.

Transaction name
The name of the IMS transaction.

Total Transactions
Total number of transactions processed by a PSB.

Program CPU time
The total dependent region CPU seconds.

DC calls GN
The total number of DL/I message queue GN calls.

DC calls GU
The total number of DL/I message queue GU calls.

DC calls ISRT
The total number of DL/I message queue ISRT calls.

DB calls DLET
The total number of DL/I database DLET calls.

DB calls GHN
The total number of DL/I database GHN calls.

DB calls GHNP
The total number of DL/I database GHNP calls.

DB calls GHU
The total number of DL/I database GHU calls.

DB calls GN
The total number of DL/I database GN calls.

DB calls GNP
The total number of DL/I database GNP calls.

DB calls GU
The total number of DL/I database GU calls.

DB calls ISRT
The total number of DL/I database ISRT calls.

DB calls REPL
The total number of DL/I database REPL calls.

IMS HALDB OLR reports

IMS HALDB OLR reports show information about the High Availability Large Databases Online Reorganization (HALDB OLR) process.

IMS HALDB OLR Unit Of Reorganization, Details

This report shows details about High Availability Large Databases Online Reorganization (HALDB OLR) activity at Unit of Reorganization (UOR) level.

This information identifies the report:

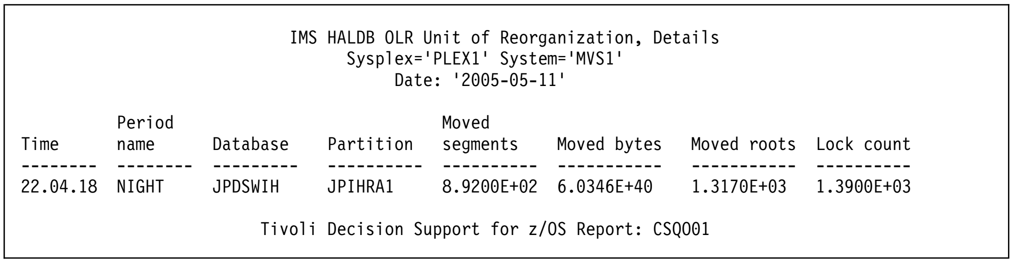
Report ID
CSQO01

Report group
CSQ reports

Source
IMS_HALDB_OLR_T

Attributes
IMS, HALDB, OLR, UOR

Variables
EVENT_DATE, PERIOD_NAME, SYSPLEX_NAME, MVS_SYSTEM_ID, DATABASE, PARTITION



```
                            IMS HALDB OLR Unit of Reorganization, Details
                                  Sysplex='PLEX1' System='MVS1'
                                       Date: '2005-05-11'

          Period                          Moved
Time      name      Database    Partition   segments    Moved bytes   Moved roots  Lock count
--------  --------  ---------   ----------  ----------  -----------   -----------  ----------
22.04.18  NIGHT     JPDSWIH     JPIHRA1     8.9200E+02  6.0346E+40    1.3170E+03   1.3900E+03

                         Tivoli Decision Support for z/OS Report: CSQO01
```

Figure 40. Example of an IMS HALDB OLR Unit of Reorganization, Details report

The report contains this information:

Time Time of the measurement. From TIME.

Period name
Period name. From PERIOD_NAME.

Database
Database. From DBD_NAME.

Partition
Database partition name. From PARTITION_NAME.

Moved segments
The number of moved segments. From SEG_MOVED.

Moved bytes
The number of moved bytes. From SIZE_MOVED.

Moved roots
The number of moved roots. From ROOTS_MOVED.

Lock count
The count of locks. From LOCK_COUNT.

IMS HALDB OLR Statistics

This report shows details about High Availability Large Databases Online Reorganization (HALDB OLR) activity.

This information identifies the report:

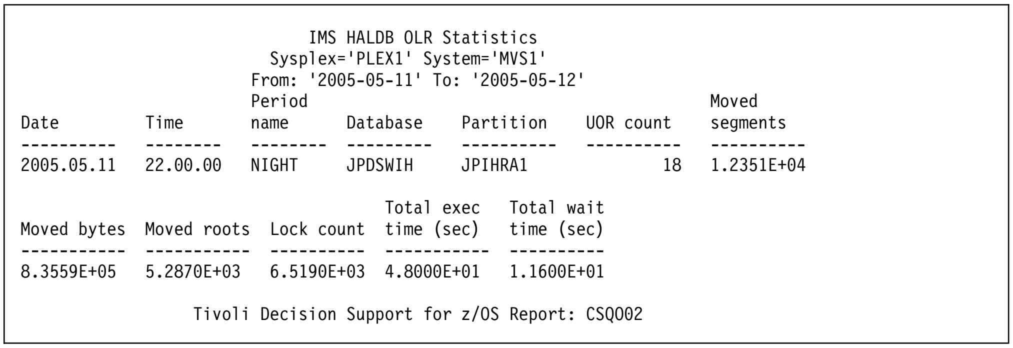
Report ID
CSQO02

Report group
CSQ reports

Source
IMS_HALDB_OLR_H

Attributes
IMS, HALDB, OLR

Variables
FROM_DATE, TO_DATE, PERIOD_NAME, SYSPLEX_NAME, MVS_SYSTEM_ID, DATABASE, PARTITION



```
                             IMS HALDB OLR Statistics
                          Sysplex='PLEX1' System='MVS1'
                        From: '2005-05-11' To: '2005-05-12'
                        Period                                       Moved
Date         Time       name      Database    Partition    UOR count    segments
----------   --------   --------  ---------   ----------   ----------   ----------
2005.05.11   22.00.00   NIGHT     JPDSWIH     JPIHRA1              18   1.2351E+04

                                       Total exec   Total wait
Moved bytes  Moved roots  Lock count  time (sec)   time (sec)
-----------  -----------  ----------  -----------  ----------
8.3559E+05   5.2870E+03   6.5190E+03  4.8000E+01   1.1600E+01

                    Tivoli Decision Support for z/OS Report: CSQ002
```

Figure 41. Example of an IMS HALDB OLR Statistics report

The report contains this information:

Date Date of the measurement. From DATE.

Time Hour of the measurement. From TIME.

Period name
Period name. From PERIOD_NAME.

Database
Database. From DBD_NAME.

Partition
Database partition name. From PARTITION_NAME.

UOR count
The number of UORs. From UOR_COUNT.

Moved segments
The number of moved segments. From SEG_MOVED.

Moved bytes
The number of moved bytes. From SIZE_MOVED.

Moved roots
The number of moved roots. From ROOTS_MOVED.

Lock count
The count of locks. From LOCK_COUNT.

Total exec time (sec)
Total execution time, in seconds. From EXEC_TIME.

Total wait time (sec)
Total wait time, in seconds. From WAIT_TIME.

Key Performance Metrics IMS reports

These reports show Processing Times by Transaction by Hour.

KPM IMS Processing Times by Transaction by Hour report

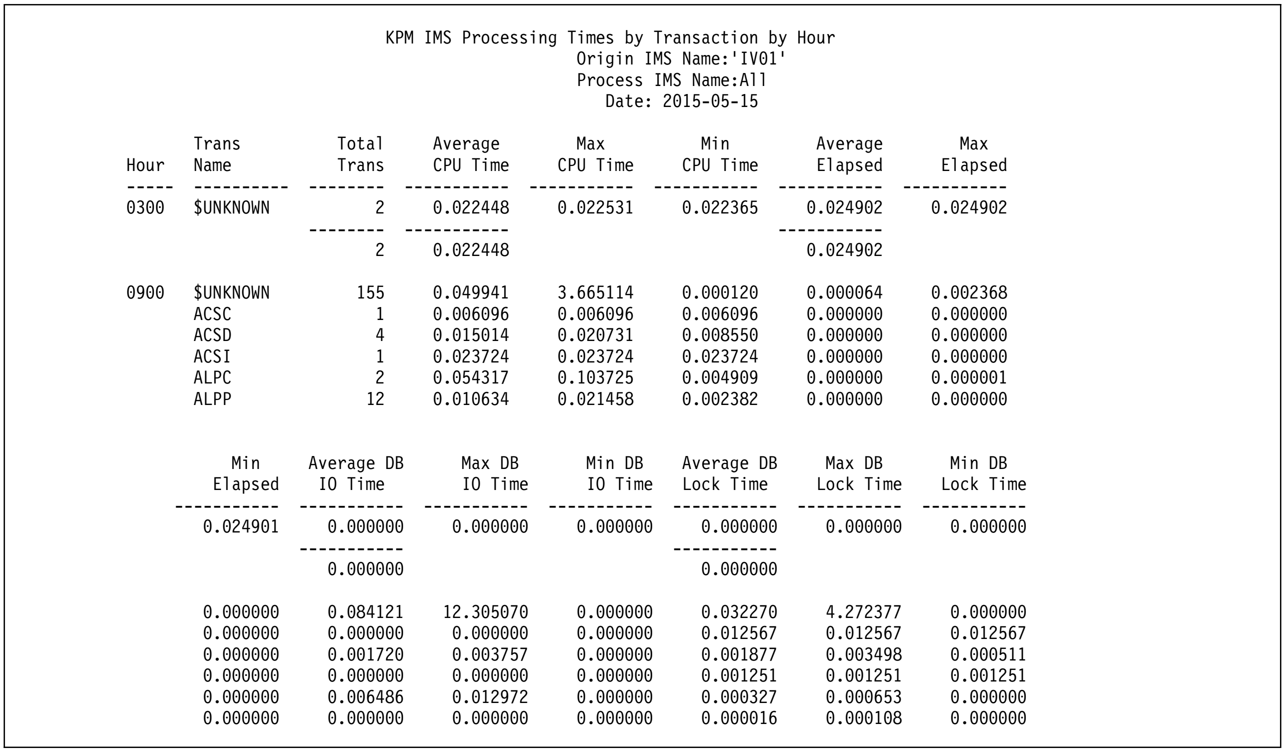
This information identifies the report:

Report ID
CSQTL01

Report Group
KPM IMS reports

Source
KPM_IMS_TRAN_H

Attributes
IMS, Transactions, Utilization, Performance, Hourly
Variables
ORIGIN_IMS, PROCESS_IMS, DATE



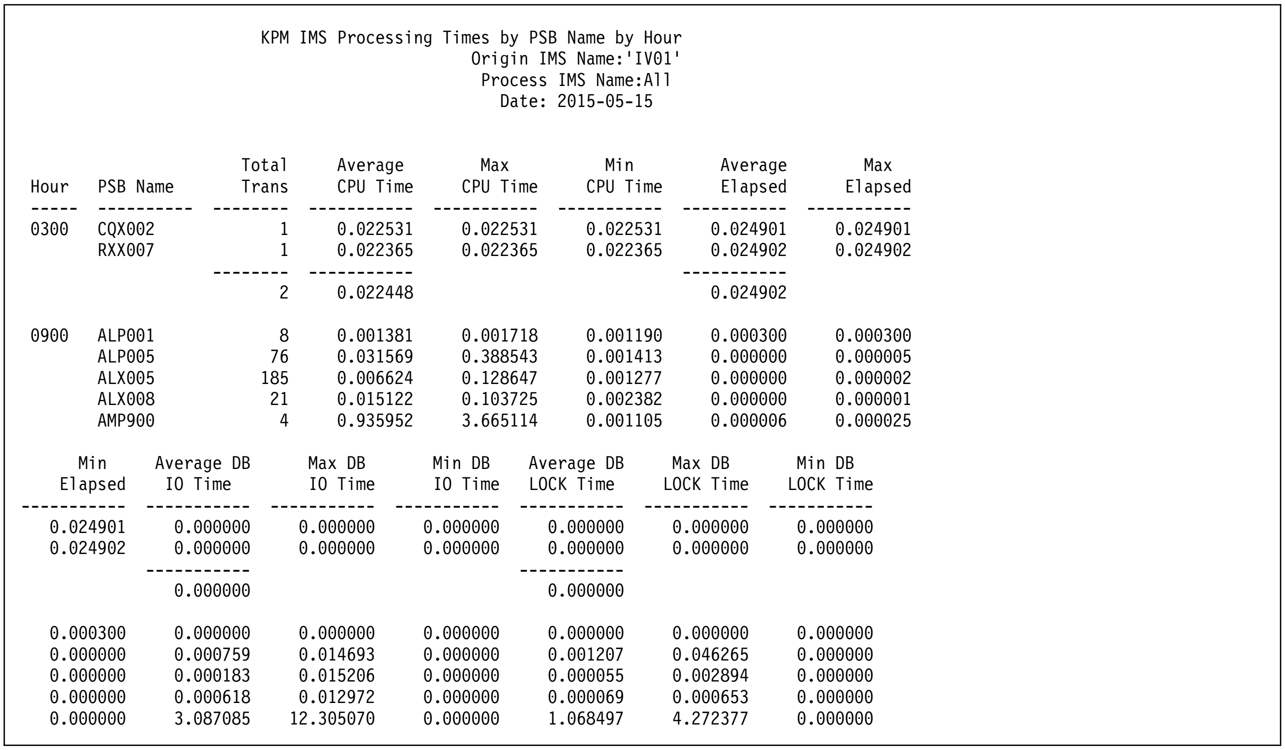
```
                           KPM IMS Processing Times by Transaction by Hour
                                               Origin IMS Name:'IV01'
                                               Process IMS Name:All
                                                  Date: 2015-05-15

       Trans          Total     Average        Max          Min         Average        Max
Hour   Name           Trans     CPU Time     CPU Time     CPU Time      Elapsed      Elapsed
-----  ----------  --------  -----------  -----------  -----------  -----------  -----------
0300   $UNKNOWN           2     0.022448     0.022531     0.022365     0.024902     0.024902
                   --------  -----------                            -----------
                          2     0.022448                               0.024902

0900   $UNKNOWN         155     0.049941     3.665114     0.000120     0.000064     0.002368
       ACSC               1     0.006096     0.006096     0.006096     0.000000     0.000000
       ACSD               4     0.015014     0.020731     0.008550     0.000000     0.000000
       ACSI               1     0.023724     0.023724     0.023724     0.000000     0.000000
       ALPC               2     0.054317     0.103725     0.004909     0.000000     0.000001
       ALPP              12     0.010634     0.021458     0.002382     0.000000     0.000000


           Min     Average DB      Max DB       Min DB    Average DB     Max DB       Min DB
         Elapsed    IO Time        IO Time      IO Time   Lock Time     Lock Time    Lock Time
     -----------  -----------  -----------  -----------  -----------  -----------  -----------
        0.024901     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
                  -----------                            -----------
                     0.000000                               0.000000

        0.000000     0.084121    12.305070     0.000000     0.032270     4.272377     0.000000
        0.000000     0.000000     0.000000     0.000000     0.012567     0.012567     0.012567
        0.000000     0.001720     0.003757     0.000000     0.001877     0.003498     0.000511
        0.000000     0.000000     0.000000     0.000000     0.001251     0.001251     0.001251
        0.000000     0.006486     0.012972     0.000000     0.000327     0.000653     0.000000
        0.000000     0.000000     0.000000     0.000000     0.000016     0.000108     0.000000
```

Figure 42. Example of a KPM IMS Processing Times by Transaction by Hour report

KPM IMS Processing Times by PSB Name by Hour report

This information identifies the report:

Report ID
CSQTL02
Report Group
KPM IMS reports
Source
KPM_IMS_TRAN_H
Attributes
IMS, PSB, Utilization, Performance, Hourly
Variables
ORIGIN_IMS, PROCESS_IMS, DATE



```
                         KPM IMS Processing Times by PSB Name by Hour
                                             Origin IMS Name:'IV01'
                                              Process IMS Name:All
                                                Date: 2015-05-15


                       Total     Average       Max         Min         Average       Max
 Hour   PSB Name       Trans     CPU Time    CPU Time    CPU Time      Elapsed     Elapsed
 -----  ----------  --------  -----------  -----------  -----------  -----------  -----------
 0300   CQX002             1     0.022531     0.022531     0.022531     0.024901     0.024901
        RXX007             1     0.022365     0.022365     0.022365     0.024902     0.024902
                    --------  -----------                            -----------
                           2     0.022448                               0.024902

 0900   ALP001             8     0.001381     0.001718     0.001190     0.000300     0.000300
        ALP005            76     0.031569     0.388543     0.001413     0.000000     0.000005
        ALX005           185     0.006624     0.128647     0.001277     0.000000     0.000002
        ALX008            21     0.015122     0.103725     0.002382     0.000000     0.000001
        AMP900             4     0.935952     3.665114     0.001105     0.000006     0.000025

     Min      Average DB      Max DB       Min DB    Average DB     Max DB      Min DB
   Elapsed    IO Time        IO Time      IO Time   LOCK Time     LOCK Time    LOCK Time
-----------  -----------  -----------  -----------  -----------  -----------  -----------
   0.024901     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
   0.024902     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
             -----------                            -----------
                0.000000                               0.000000

   0.000300     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
   0.000000     0.000759     0.014693     0.000000     0.001207     0.046265     0.000000
   0.000000     0.000183     0.015206     0.000000     0.000055     0.002894     0.000000
   0.000000     0.000618     0.012972     0.000000     0.000069     0.000653     0.000000
   0.000000     3.087085    12.305070     0.000000     1.068497     4.272377     0.000000
```

Figure 43. Example of a KPM IMS Processing Times by PSB Name by Hour report

KPM IMS Processing Times by Region Type by Hour report

This information identifies the report:

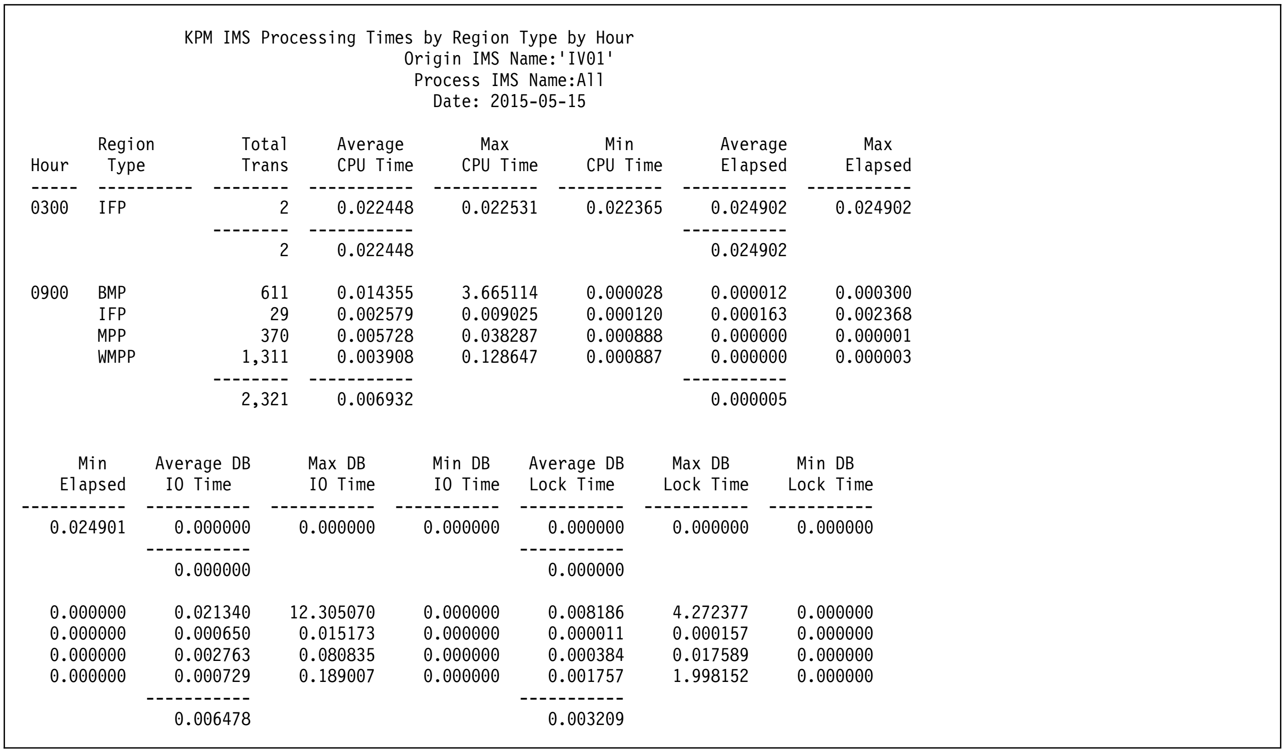
Report ID
CSQTL03

Report Group
KPM IMS reports

Source
KPM_IMS_TRAN_H

Attributes
Attributes

Variables
ORIGIN_IMS, PROCESS_IMS, DATE



```
                  KPM IMS Processing Times by Region Type by Hour
                                       Origin IMS Name:'IV01'
                                        Process IMS Name:All
                                          Date: 2015-05-15

        Region         Total     Average       Max          Min         Average       Max
 Hour    Type          Trans     CPU Time    CPU Time     CPU Time     Elapsed      Elapsed
 -----  ----------  --------  -----------  -----------  -----------  -----------  -----------
 0300   IFP                2     0.022448     0.022531     0.022365     0.024902     0.024902
                    --------  -----------                             -----------
                           2     0.022448                               0.024902

 0900   BMP              611     0.014355     3.665114     0.000028     0.000012     0.000300
        IFP               29     0.002579     0.009025     0.000120     0.000163     0.002368
        MPP              370     0.005728     0.038287     0.000888     0.000000     0.000001
        WMPP           1,311     0.003908     0.128647     0.000887     0.000000     0.000003
                    --------  -----------                             -----------
                       2,321     0.006932                               0.000005


     Min       Average DB      Max DB       Min DB     Average DB     Max DB       Min DB
   Elapsed      IO Time       IO Time      IO Time     Lock Time     Lock Time    Lock Time
 -----------  -----------  -----------  -----------  -----------  -----------  -----------
    0.024901     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
              -----------                            -----------
                 0.000000                               0.000000

    0.000000     0.021340    12.305070     0.000000     0.008186     4.272377     0.000000
    0.000000     0.000650     0.015173     0.000000     0.000011     0.000157     0.000000
    0.000000     0.002763     0.080835     0.000000     0.000384     0.017589     0.000000
    0.000000     0.000729     0.189007     0.000000     0.001757     1.998152     0.000000
              -----------                            -----------
                 0.006478                               0.003209
```

Figure 44. Example of a KPM IMS Processing Times by PSB Name by Hour report

KPM IMS Average Enqueues by Transaction by Hour report

This information identifies the report:

Report ID
CSQTL04

Report Group
KPM IMS reports

Source
KPM_IMS_TRAN_H

Attributes
IMS, Enqueues, Utilization, Performance, Hourly

Variables
ORIGIN_IMS, PROCESS_IMS, DATE

```
                    KPM IMS Average Enqueues by Transaction by Hour
                                             Origin IMS Name:'IV01'
                                              Process IMS Name:All
                                               Date: 2015-05-15


                                    Ave       Ave       Ave       Ave       Ave       Ave       Ave
       Trans                Total   Test      Test      Test      Queue     Queue     Queue     Updt
 Hour  Name                 Trans  Enqueues  Enqwaits  Dequeues  Enqueues  Enqwaits  Dequeues  Enqueues
 ----- ----------  --------  --------  --------  --------  --------  --------  --------  --------
 0300  $UNKNOWN             2     0.000     0.000     0.000     0.000     0.000     0.000     0.000
                   --------  --------  --------  --------  --------  --------  --------  --------
                            2     0.000     0.000     0.000     0.000     0.000     0.000     0.000

 0900  $UNKNOWN           155     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ACSC                 1     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ACSD                 4     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ACSI                 1     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ALPC                 2     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ALPP                12     0.000     0.000     0.000     0.000     0.000     0.000     0.000
       ALPQ                 4     0.000     0.000     0.000     0.000     0.000     0.000     0.000


   Ave       Ave       Ave       Ave       Ave
  Updt      Updt      Excl      Excl      Excl
Enqwaits  Dequeues  Enqueues  Enqwaits  Dequeues
--------  --------  --------  --------  --------
   0.000     0.000     0.000     0.000     0.000
--------  --------  --------  --------  --------
   0.000     0.000     0.000     0.000     0.000

   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000
```

Figure 45. Example of a KPM IMS Average Enqueues by Transaction by Hour report

KPM IMS DB Activity by Region by Program Name by Hour report

This information identifies the report:

Report ID
CSQTL05

Report Group
KPM IMS reports

Source
KPM_IMS_TRAN_H

Attributes
IMS, Data Base Activity, Program, Performance, Hourly

Variables
ORIGIN_IMS, PROCESS_IMS, DATE

```
                 KPM IMS DB Activity by Region by Program Name by Hour
                                              Origin IMS Name:'IV
                                               Process IMS Name:A
                                                 Date: 2015-05-15


                                                                    Average
        Transact  PSB       Region          Total  Approx CPU     Response    DB2 Read   DB2 Write
 Hour     Name    Name       Type           Trans  Utilization        Time     Calls       Calls
 -----  --------  --------  --------  -----------  -----------  -----------  ---------- -----------
 0300   $UNKNOWN  CQX002    IFP                 1     0.022531     0.024901           0           0
        $UNKNOWN  RXX007                        1     0.022365     0.024902           0           0
                                      -----------  -----------  -----------  ---------- -----------
                                                2     0.044896     0.024902           0           0

                                      -----------  -----------  -----------  ---------- -----------
                                                2     0.044896     0.024902           0           0

 0900   $UNKNOWN  ALP001    BMP                 8     0.011050     0.000300           0           0
        $UNKNOWN  ALP005                       76     2.399230     0.000000         223          55
                                      -----------  -----------  -----------  ---------- -----------
                                               84     2.410280     0.000029         223          55

        ALVS      ALX005    WMPP              185     1.225476     0.000000         247           0
        ALPC      ALX008                        2     0.108633     0.000000           7           0
        ALPP      ALX008                       12     0.127605     0.000000          52           0
        ALPQ      ALX008                        4     0.052412     0.000000          16           0
        ALRP      ALX008                        1     0.012439     0.000000           0           0
        ALTH      ALX008                        2     0.016463     0.000000           0           0
                                      -----------  -----------  -----------  ---------- -----------
                                              206     1.543029     0.000000         322           0

 DB2 Delete    DB2 Total
   Calls         Calls     VSAM Reads  VSAM Writes   OSAM Reads  OSAM Writes    Total IOS
-----------  -----------  -----------  -----------  -----------  -----------  -----------
          0            0            0            0            0            0            0
          0            0            0            0            0            0            0
-----------  -----------  -----------  -----------  -----------  -----------  -----------
          0            0            0            0            0            0            0

-----------  -----------  -----------  -----------  -----------  -----------  -----------
          0            0            0            0            0            0            0

          0            0            0            0            0            0            0
          0          278           14            0            0            0           14
-----------  -----------  -----------  -----------  -----------  -----------  -----------
          0          278           14            0            0            0           14

          0          247           16            0            0            0           16
          0            7            3            0            0            0            3
          0           52            0            0            0            0            0
          0           16            0            0            0            0            0
          0            0            0            0            0            0            0
          0            0            0            0            0            0            0
-----------  -----------  -----------  -----------  -----------  -----------  -----------
          0          322           19            0            0            0           19
```

Figure 46. Example of a KPM IMS DB Activity by Region by Program Name by Hour report

Appendix B. Creating IMS log record DSECTs

This appendix can help you create log record dummy control sections (DSECTs) for use with the IMS Performance feature. Figure 47 on page 152 shows an example of a job that you can use to assemble the log records mapping macro ILOGREC that is provided with IMS. The sample shown is the most inclusive method of extracting DSECTs. You can modify it to suit your own needs. You can write your own record procedures that use the composite record created by the IMS Performance feature.

Creating IMS log record DSECTs

```
//USERIDA  JOB (ACC000,001),'ILOGREC',
//         NOTIFY=USERID,MSGCLASS=A,CLASS=A,REGION=0M
//DSECT    PROC
//HASM     EXEC PGM=IEV90,PARM='NODECK,NOXREF,LIST,NORLD,NOOBJECT'
//SYSLIB   DD  DISP=SHR,DSN=IMS71.GENLIBB,DCB=SYS1.MACLIB
//         DD  DISP=SHR,DSN=IMS71.GENLIBA
//         DD  DISP=SHR,DSN=IMS71.GENLIB
//         DD  DISP=SHR,DSN=SYS1.MACLIB
//SYSUT1   DD  UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT2   DD  UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSUT3   DD  UNIT=SYSDA,SPACE=(CYL,(2,1))
//SYSPRINT DD  SYSOUT=*
//         PEND
//D1       EXEC DSECT              /* Full function record DSECTs */
//SYSIN    DD  *
  ILOGREC  RECID=ALL
           END
/*
//D2       EXEC DSECT              /* Fast path record DSECTs     */
//SYSIN    DD  *
  DBFLGRIM
           EJECT
  DBFLGROM
           EJECT
  DBFBMSDB
           EJECT
  DBFDOCL
           EJECT
  DBFLGRDQ
           EJECT
  DBFLGSYN
           EJECT
  DBFLGRIC
           EJECT
  DBFLSRT  RECID=50
           EJECT
  DBFLGRSD
           EJECT
  DBFLGRRE
           END
/*
//D3       EXEC DSECT              /* IMS control block DSECTs    */
//SYSIN    DD  *
  IBPOOL
           EJECT
  IDLIVSAM BFSP
           EJECT
  ICLI     CTTBASE=1
           EJECT
  ICLI     CNTBASE=1
           EJECT
  ICLI     SPQBASE=1
           EJECT
  ICLI     CTBBASE=1
           EJECT
  ICLI     CIBBASE=1

           EJECT
  ICLI     PCIBASE=1
           EJECT
  ICLI     CRBBASE=1
           EJECT
  ICLI     CLBBASE=1
           EJECT
  ICLI     CVBBASE=1
           EJECT
  ICLI     CCBBASE=1
           EJECT
  ICLI     BUFBASE=1
           EJECT
  ICLI     POOBASE=1
           EJECT
  ICLI     BFRBASE=1
           EJECT
```

Appendix C. Sample archive exit

This appendix shows a description from the sample IMS archive exit, DRLJXIMS. The source for this sample exit can also be found in member DRLJXIMS in the product CNTL library. Figure 48 on page 156 shows the sample exit.

To assemble this exit, you need access to both the MVS and the IMS macro libraries. Refer to your IMS system documentation for detailed information regarding the archive exit.

```
         TITLE 'DRLJXIMS - IMS ARCHIVE EXIT'
DRLJXIMS CSECT
         SPACE
***********************************************************************
*                                                                     *
*  MODULE NAME:  DRLJXIMS                                             *
*                                                                     *
*  DESCRIPTION:  IMS ARCHIVE EXIT FOR EPDM/IMS                        *
*                                                                     *
*  COPYRIGHT:    NONE                                                 *
*                                                                     *
*  STATUS:       IMS/ESA V3R1                                         *
*                                                                     *
*  FUNCTION:                                                          *
*    WRITES THE RECORDS USED BY EPDM/IMS TO THE FILE DEFINED BY THE   *
*    DDNAME IMSLOG.                                                   *
*                                                                     *
*    THE RECORD TYPES CONCERNED ARE AS FOLLOWS (ALL IN HEX)           *
*                                                                     *
*       01, 02, 03, 06, 07, 08, 0A, 11, 12, 13, 16, 24,               *
*       31, 32, 33, 34, 35, 36, 37, 38, 42, 45,                       *
*       47, 55 AND 56                                                 *
*       40, SUBTYPES 01, 04, AND 98.                                  *
*       59, SUBTYPES 01, 03, 36, 37, AND 38.                          *
*                                                                     *
*    THIS MINIMIZES THE AMOUNT OF DATA PASSED TO EPDM/IMS.            *
*                                                                     *
*  LOGIC:                                                             *
*    CASE INIT (DRLJXIMS CALL CODE 1).                                *
*      GETMAIN STORAGE FOR WORK AREAS AND ANCHOR IT IN THE USER       *
*      WORD.                                                          *
*      OPEN OUTPUT FILE.                                              *
*    END CASE INIT.                                                   *
*                                                                     *
*    CASE NORMAL (DRLJXIMS CALL CODE 2).                              *
*                                                                     *
*      SUBCASE RECORD TYPES                                           *
*                           01, 02, 03, 06, 07, 08, 0A, 11, 12, 13,   *
*                           16, 24, 31, 32, 33, 34, 35, 36, 37, 38,   *
*                           42, 45, 47, 55 AND 56                     *
*        COPY RECORD.                                                 *
*      END SUBCASE RECORD TYPES                                       *
*                           01, 02, 03, 06, 07, 08, 0A, 11, 12, 13,   *
*                           16, 24, 31, 32, 33, 34, 35, 36, 37, 38,   *
*                           42, 45, 47, 55 AND 56                     *
*                                                                     *
*      SUBCASE RECORD TYPE                                            *
*                           40, SUBTYPES 01, 04, AND 98.              *
*        COPY RECORD.                                                 *
*      END SUBCASE RECORD TYPE                                        *
*                           40, SUBTYPES 01, 04, AND 98.              *
*                                                                     *
*      SUBCASE RECORD TYPE                                            *
*                           59, SUBTYPES 01, 03, 36, 37, AND 38.      *
*        COPY RECORD.                                                 *
*      END SUBCASE RECORD TYPE                                        *
*                           59, SUBTYPES 01, 03, 36, 37, AND 38.      *
*                                                                     *
*    END CASE NORMAL.                                                 *
*                                                                     *
*    CASE TERMINATE (DRLJXIMS CALL CODE 3).                           *
*      CLOSE OUTPUT FILE.                                             *
*      FREEMAIN STORAGE FOR WORK AREAS AND RESET ANCHOR POINTER.      *
*    END CASE TERMINATE.                                              *
```

Figure 48. Description from the sample IMS archive exit

Appendix D. DFSLTMG0 log merge utility

The DFSLTMG0 log merge utility produces one data set by merging the system log data sets (SLDS) from two or more IMS systems. The log merge utility can merge up to nine IMS system logs from the same IMS release. Each log is the output of a uniquely identified IMS system running during the same time span. The order of input to the log merge utility is LOG01, LOG02... LOG09. DFSLTMG0 is placed in IMS.RESLIB during IMS system definition.

```
//STEP0 EXEC PGM=DFSLTMG0
//STEPLIB   DD DSNAME=IMS.RESLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//PRINT     DD SYSOUT=*
//LOG01     DD DSNAME=xxx.IMSLOG01,DISP=OLD
//LOG02     DD DSNAME=xxx.IMSLOG02,DISP=OLD
//LOGOUT    DD DSNAME=xxx.IMSLOG.MERGE,DISP=(NEW,CATLG)
//            VOL=SER=yyyy,UNIT=3390,SPACE=(CYL,(100,5)),
//            DCB==(RECFM=VB,LRECL=27994,BLKSIZE=32760)
//SYSIN     DD  *
START 01157,0001
STOP  01157,2359
MSG
//*
```

Figure 49. DFSLTMG0 log merge utility

Following is the description of the statements:

STEPLIB DD

This statement points to IMS.RESLIB, which contains the IMS nucleus and required action modules.

```
//STEPLIB DD DSNAME=IMS.RESLIB,DISP=SHR
```

PRINT DD

Indicates the SYSPRINT data set used for control statements and error messages.

```
//PRINT DD SYSOUT=A
```

LOG01 DD

Describes the first input log data set.

```
//LOG01 DD DSNAME=IMS.LOGA,DISP=OLD,
// VOL=SER=XXXXXX,UNIT=TAPE
```

LOG02 DD

Describes the second input log data set.

```
//LOG02 DD DSNAME=IMS.LOGB,DISP=OLD,
// VOL=SER=XXXXXX,UNIT=TAPE
```

LOGOUT DD

Describes the output data set.

```
//LOGOUT DD DSNAME=IMS.LOGOUT,DISP=(,PASS),
// VOL=SER=YYYYYY,UNIT=TAPE,
// DCB=(RECFM=VBS,LRECL=6000,BLKSIZE=6008)
```

SYSIN DD

Describes the control statement data set.

```
//SYSIN DD *
```

START
Used to specify a start time. This statement must be present (yyddd, hhmmsstt).

```
START
```

STOP You must specify a stop time, which must be relative to the time field in LOG01 (yyddd, hhmmsstt)

```
STOP
```

Log Record Selection
Use this control statement to merge only certain types of log records.

Controlling the log merge

About this task

To control the log output, you need to:

Procedure

1. Choose the required systems to participate in the logical link paths you want to examine.
2. Coordinate the series of input logs for each system so that they cover a similar time span.
3. Specify a start and stop time for the Log Merge utility control statements if you want to sample the cross-system processing for a particular interval.
4. Specify MSG to select log records that are suitable for the transaction analysis step. Records is the default, but this means the DL/I activity for several systems is included in the utility input, and this can cause extended processing time.

Appendix E. Tailoring example using MSGTEXT for IMS_TRAN_x tables

About this task

The **MSGTEXT** log procedure parameter allows you to extract up to 60 bytes of user data from the MSGXSTXT field of type X'01' and type X'03' log records and store that data in the CQS feature transaction data tables. To store the data, you must customize at least some of your record, update, and table definitions. This appendix provides an example customization. For more information about specifying the **MSGTEXT** parameter see "Specifying log procedure parameters" on page 33.

To store the extracted data into DB2 you will need to customize your IMS feature tables. The following instructions describe how to store the data in the IMS_TRAN_x tables. Similar changes would be required for the other transaction-oriented tables. You can give the column any name you like, but you should be consistent.

This example covers IMS V12 only (additional members would need to be customized to support other versions of IMS). In this example the DRLIPARM parameter **MSGTEXT=1,10** is used to extract 10 bytes of user data from the MSGXSTXT field of the IMS X'01' and X'03' records. The new 10 byte field will be a new key field in the database. These are the DB2 table definition changes required for the IMS_TRAN_x tables.

Copy the members indicated to your LOCAL.DEFS data set and perform the customization as indicated:

Procedure

1. Customize SDRLDEFS(DRLTCSQR). This contains the definitions of the IMS_TRAN_x tables and indexes.
   a. Add `USER_DATA CHAR(10) NOT NULL` to the list of key columns in each of the table definitions following the TRANSACTION_NAME column.
   b. Add `USER_DATA` to the primary key definition in each of the table definitions and to the list of key columns in the index definitions.
2. Customize SDRLDEFS(DRLRSC1C). This contains the CSQ_VC10_R2 and CSQ_VC10_R2_LIGHT record definitions.
   a. At the start of the definitions for CSQ_VC10_R2 and CSQ_VC10_R2_LIGHT, change the string in the VERSION statement following the DEFINE to a unique string for your modification.
   b. At the start of the definition for CSQ_VC10_R2_LIGHT, either remove the line`IDENTIFIED BY DRL_LENGTH = 246`, or change it to include the new record length `IDENTIFIED BY DRL_LENGTH = 246 OR DRL_LENGTH = `***nnn***, where ***nnn*** is the record length of the extended record containing the USER_DATA. In this example ***nnn*** will be 258 (246 + 2 (length prefix) + 10 (user-data)).
   c. At the end of the definitions for CSQ_VC10_R2 and CSQ_VC10_R2_LIGHT replace the following:

```
DRL_TXCLS   LENGTH 2 HEX);             -- Transaction class
```

   with

```
DRL_TXCLS   LENGTH 2 HEX,             -- Transaction class
DRL_MSGTXL  LENGTH 2 BINARY,          -- User data length
DRL_MSGTXT  CHAR(10));                -- User data
```

3. Customize SDRLDEFS(DRLUIC1C). This contains the CSQVC10TRANH and CSQVC10TRANLH update definitions. At the start of the definitions:
   a. Change the string in the VERSION statement following the DEFINE to a unique string for your modification.
   b. In the LET clause for each definition following TRANSACTION_NAME add:
      ```
      W_USERDATA   = CASE DRL_MSGTXL
                      WHEN 10 THEN DRL_MSGTXT
                      ELSE '$BLANK  '
                        END
      ```
   c. Following the TRANSACTION_NAME assignment in the GROUP BY clause for each definition add:
      ```
      USER_DATA   = W_USERDATA  ,
      ```
4. Customize SDRLDEFS(DRLUIMSC). This contains the IMS_TRAN_D, IMS_TRAN_H, and IMS_TRAN_W update definitions. At the start of the definitions:
   - Change the string in the VERSION statement following the DEFINE to a unique string for your modification.
   - In the GROUP BY clause for each definition following the TRANSACTION_NAME assignment add:
     ```
     USER_DATA   = USER_DATA  ,
     ```
5. Install or Reinstall the Component to activate the local modifications.
   a. If the IMS 12.1 CSQ Collect Component has not been installed previously:
      - Install the IMS 12.1 CSQ Collect Component using the new definitions created in your LOCAL.DEFS data set.
      - The IMS_TRAN_H, IMS_TRAN_D, and IMS_TRAN_W tables will contain the new definition of your USER_DATA field.
   b. If the IMS 12.1 CSQ Collect Component is already installed:
      - UNLOAD the data you have stored in the IMS_TRAN_H, IMS_TRAN_D, and IMS_TRAN_W tables.
      - DROP the IMS_TRAN_H, IMS_TRAN_D, and IMS_TRAN_W tables.
      - Reinstall the IMS 12.1 CSQ Collect Component using the new definitions stored in your LOCAL.DEFS data set.
      - The IMS_TRAN_H, IMS_TRAN_D, and IMS_TRAN_W tables will be recreated and now contain the new definition of your USER_DATA field.
      - Re-LOAD your data to the IMS_TRAN_H, IMS_TRAN_D, and IMS_TRAN_W tables. You will need to add the new field names to the UNLOAD SYSPUNCH control statements created by the UNLOAD, these are used as SYSIN to re-LOAD the tables.

Accessing the extracted information

About this task

By specifying a valid **MSGTEXT** parameter, a length-prefixed data field is appended to the IMS feature transaction (or R2) record produced when the IMS log is processed. You can use the DRLSLOGP program to produce a file containing the R2 records for review.

Procedure

In your DRLSLOGP JCL specify a DRLIRPT2 data set with a RECFM=F or RECFM=V, with an LRECL of at least 246 + the length from the **MSGTEXT** parameter, plus two bytes for the user data length prefix, plus the length from the MSGTEXT parameter you coded in your DRLIPARM.
Where data is available, in the R2 record, you will see the 2 byte hex length at column 247, followed by the selected data starting at column 249.

Appendix F. Support information

If you have a problem with your IBM software, you want to resolve it quickly. This section describes the following options for obtaining support for IBM software products:

- "Searching knowledge bases"
- "Obtaining fixes"
- "Receiving weekly support updates" on page 164
- "Contacting IBM Software Support" on page 164

Searching knowledge bases

You can search the available knowledge bases to determine whether your problem was already encountered and is already documented.

Searching the information center

IBM provides extensive documentation that can be installed on your local computer or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, and reference information.

Searching the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem.

To search multiple Internet resources for your product, use the **Web search** topic in your information center. In the navigation frame, click **Troubleshooting and support** ▸ **Searching knowledge bases** and select **Web search**. From this topic, you can search a variety of resources, including the following:

- IBM technotes
- IBM downloads
- IBM developerWorks®
- Forums and newsgroups
- Google

Obtaining fixes

A product fix might be available to resolve your problem. To determine what fixes are available for your IBM software product, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/software/support/.
2. Click **Downloads and drivers** in the **Support topics** section.
3. Select the **Software** category.
4. Select a product in the **Sub-category** list.
5. In the **Find downloads and drivers by product** section, select one software category from the **Category** list.
6. Select one product from the **Sub-category** list.

7. Type more search terms in the **Search within results** if you want to refine your search.
8. Click **Search**.
9. From the list of downloads returned by your search, click the name of a fix to read the description of the fix and to optionally download the fix.

For more information about the types of fixes that are available, see the *IBM Software Support Handbook* at http://www-304.ibm.com/support/customercare/sas/f/handbook/home.html.

Receiving weekly support updates

To receive weekly e-mail notifications about fixes and other software support news, follow these steps:

1. Go to the IBM Software Support Web site at http://www.ibm.com/support/us/.
2. Click **My support** in the upper right corner of the page.
3. If you have already registered for **My support**, sign in and skip to the next step. If you have not registered, click **register now**. Complete the registration form using your e-mail address as your IBM ID and click **Submit**.
4. Click **Edit profile**.
5. In the **Products** list, select **Software**. A second list is displayed.
6. In the second list, select a product segment, for example, **Application servers**. A third list is displayed.
7. In the third list, select a product sub-segment, for example, **Distributed Application & Web Servers**. A list of applicable products is displayed.
8. Select the products for which you want to receive updates, for example, **IBM HTTP Server** and **WebSphere®** **Application Server**.
9. Click **Add products**.
10. After selecting all products that are of interest to you, click **Subscribe to email** on the **Edit profile** tab.
11. Select **Please send these documents by weekly email**.
12. Update your e-mail address as needed.
13. In the **Documents** list, select **Software**.
14. Select the types of documents that you want to receive information about.
15. Click **Update**.

If you experience problems with the **My support** feature, you can obtain help in one of the following ways:

Online
Send an e-mail message to erchelp@ca.ibm.com, describing your problem.

By phone
Call 1-800-IBM-4You (1-800-426-4968).

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.

Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2 and WebSphere products that run on Windows, or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:

  Online
  Go to the Passport Advantage Web site at http://www.lotus.com/services/passport.nsf/ WebDocs/Passport_Advantage_Home and click **How to Enroll**.

  By phone
  For the phone number to call in your country, go to the IBM Software Support Web site at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.
- For customers with Subscription and Support (S & S) contracts, go to the Software Service Request Web site at https://techsupport.services.ibm.com/ssr/login.
- For customers with IBMLink, CATIA, Linux, S/390®, iSeries, pSeries, zSeries, and other support agreements, go to the IBM Support Line Web site at http://www.ibm.com/services/us/index.wss/so/its/a1000030/dt006.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries, pSeries, and iSeries environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web site at http://www.ibm.com/servers/eserver/techsupport.html.

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States. From other countries, go to the contacts page of the *IBM Software Support Handbook* on the Web at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region for phone numbers of people who provide support for your location.

To contact IBM Software support, follow these steps:

1. "Determining the business impact"
2. "Describing problems and gathering information" on page 166
3. "Submitting problems" on page 166

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2

The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3

The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4

The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing problems and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting problems

You can submit your problem to IBM Software Support in one of two ways:

Online

Click **Submit and track problems** on the IBM Software Support site at http://www.ibm.com/software/support/probsub.html. Type your information into the appropriate problem submission form.

By phone

For the phone number to call in your country, go to the contacts page of the *IBM Software Support Handbook* at http://techsupport.services.ibm.com/guides/contacts.html and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the Software Support Web site daily, so that other users who experience the same problem can benefit from the same resolution.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to

IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Bibliography

Tivoli Decision Support publications

Administration Guide and Reference, SH19-6816

Resource Accounting for z/OS, SH19-4495

AS/400 System Performance Feature Guide and Reference, SH19-4019

CICS Performance Feature Guide and Reference, SH19-6820

Distributed Systems Performance Feature Guide and Reference, SH19-4018

Guide to Reporting, SH19-6842

IMS Performance Feature Guide and Reference, SH19-6825

Language Guide and Reference, SH19-6817

Messages and Problem Determination, SH19-6902

Network Performance Feature Installation and Administration, SH19-6901

Network Performance Feature Reference, SH19-6822

Network Performance Feature Reports, SH19-6821

Resource Accounting, SH19-6818

System Performance Feature Guide, SH19-6819

System Performance Feature Reference Volume I, SH19-4494

System Performance Feature Reference Volume II, SC23-7966

Glossary

administration
A Tivoli Decision Support for z/OS task that includes maintaining the database, updating environment information, and ensuring the accuracy of data collected.

administration dialog
A set of host windows used to administer Tivoli Decision Support for z/OS.

collect A process used by Tivoli Decision Support for z/OS to read data from input log data sets, interpret records in the data set, and store the data in DB2 tables in the Tivoli Decision Support for z/OS database.

component
An optionally installable part of a Tivoli Decision Support for z/OS feature. Specifically in Tivoli Decision Support for z/OS, a component refers to a logical group of objects used to collect log data from a specific source, to update the Tivoli Decision Support for z/OS database using that data, and to create reports from data in the database.

control table
A predefined Tivoli Decision Support for z/OS table that controls results returned by some log collector functions.

data table
A Tivoli Decision Support for z/OS table that contains performance data used to create reports.

environment information
All of the information that is added to the log data to create reports. This information can include data such as performance groups, shift periods, installation definitions, and so on.

IMS See "**Information Management System**".

Information Management System
A transactional and hierarchical database management system for critical on-line operational and on-demand business applications and data, enabling information integration, management, and scalability.

internal data type
A data type used within Tivoli Decision Support for z/OS during the processing of data.

key columns
The columns of a DB2 table that together constitute the key.

key value
Value that is used to sort records into groups.

log collector
A Tivoli Decision Support for z/OS program that processes log data sets and provides other TTivoli Decision Support for z/OS services.

log collector language
Tivoli Decision Support for z/OS statements used to supply definitions to and invoke services of the log collector.

log data set
Any sequential data set that is used as input to Tivoli Decision Support for z/OS.

log definition
The description of a log data set processed by the log collector.

log procedure
A program module that is used to process all record types in certain log data sets.

lookup expression
An expression that specifies how a value is obtained from a lookup table.

lookup table
A Tivoli Decision Support for z/OS DB2 table that contains grouping, translation, or substitution information.

PCB See **Program Control Block**.

Program Control Block
A control block that contains pointers to IMS databases.

purge condition
Instruction for purging old data from the Tivoli Decision Support for z/OS database.

Program Specification Block
A control block that identifies the destinations and databases used by the application program. A PSB consists of one or more program communication blocks (PCBs).

PSB See **Program Specification Block**.

record definition
The description of a record type contained in the log data sets used by Tivoli Decision Support for z/OS, including detailed record layout and data formats.

record procedure
A program module that is called to process some types of log records.

record type
The classification of records in a log data set.

repeated section
A section of a record that occurs more than once, with each occurrence adjacent to the previous one.

report definition language
Tivoli Decision Support for z/OSstatements used to define reports and report groups.

report group
A collection of Tivoli Decision Support for z/OS reports that can be referred to by a single name.

reporting dialog
A set of host or workstation windows used to request reports.

resource group
A collection of network resources that are identified as belonging to a particular department or division. Resources are organized into groups to reflect the structure of an organization.

resource information
Environment information that describes the elements in a network.

section
A structure within a record that contains one or more fields and may contain other sections.

source In an update definition, the record or DB2 table that contains the data used to update a Tivoli Decision Support for z/OS DB2 table.

sysplex
A set of systems communicating and cooperating with each other, through certain multisystem hardware components and software services, in order to process customer workloads.

system table
A DB2 table that stores information that controls log collector processing, TTivoli Decision Support for z/OS dialogs, and reporting.

target In an update definition, the DB2 table in which Tivoli Decision Support for z/OS stores data from the source record or table.

threshold
The maximum or minimum acceptable level of usage. Usage measurements are compared with threshold levels.

Tivoli Decision Support for z/OS database
A set of DB2 tables that includes data tables, lookup tables, system tables, and control tables.

update definition
Instructions for entering data into DB2 tables from records of different types or from other DB2 tables.

view An alternative representation of data from one or more tables. A view can include all or some of the columns contained in the table on which it is defined.

Index

Printed in USA

SH19-6825-13