IMS™ Performance Feature Guide and Reference

Version 16
IMS™ Performance Feature Guide and Reference

Version 16
Note
Before using this information and the product it supports, read the information in “Notices” on page 237.

7th Edition (June 2003)

This edition applies to version 1, release 6 of Tivoli Decision Support for OS/390 (product number 5695-101) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SH19-6825-05.

© Copyright International Business Machines Corporation 1993, 2003. All rights reserved.

Note to U.S. Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corporation.
Contents

Figures ........................................ vii

Tables ........................................... ix

Preface .......................................... xi
Who should read this book ....................... xi
What this book contains......................... xi
Publications ...................................... xii
Tivoli Decision Support for OS/390 library xii
Using LookAt to look up message explanations xiii
Accessing publications online .................... xiv
Ordering publications ............................. xiv
Accessibility ...................................... xv
Contacting software support ...................... xv
Conventions used in this book .................... xv
Typeface conventions ............................. xv
Changes in this edition ........................... xvi

Part 1. IMS Performance feature ........ 1

Chapter 1. Introduction to the IMS
Performance feature ......................... 3
Understanding the IMS Performance feature 3
Collecting data ................................ 4
SLDS .............................................. 4
Log procedure ................................ 4
Composite record ............................... 5
Record procedure ............................... 5
Record definitions .............................. 5
Tivoli Decision Support for OS/390 data tables and environmental information 5
Reports .......................................... 5
The log collector and DRL2LOGP .......... 5
Installing and customizing the IMS Performance feature ....................... 6
Planning for the IMS Performance feature 6
Selecting IMS Performance feature components 7
The collect components ....................... 7
The log records components ................... 8
Updating lookup tables ......................... 8
Updating the IMS_APPLICATION lookup table 8
Updating other lookup and control tables 9
Using the IMS Performance feature ........ 9

Chapter 2. Using log and record procedures within the IMS performance feature 11
The log procedure ................................ 11
Record grouping ................................ 11
Set relationships ............................... 13
Composite records and subtypes ............... 13
Handling of special IMS cases ................. 14
Release dependency ............................ 18
Log procedure DRLOUT reports ............... 18
Record procedures ............................. 19

Chapter 3. Understanding data flow through IMS performance feature 23
Overview of Tivoli Decision Support for OS/390 data flow ......................... 23
Log collector data flow ......................... 25
DRL2LOGP data flow ........................... 27
IMS Performance feature data flow .......... 28

Chapter 4. Administering the IMS Performance feature 31
Specifying DRL2LOGP and log procedure parameters ......................... 31
Specifying log collector parameters .......... 38
Running the log collector ...................... 39
Using DRL2LOGP ............................... 40
DRL2LOGP input and output data sets ..... 40
Running DRL2LOGP ......................... 41
Operational considerations .................... 42
Recovering from abends during collect ..... 43
  Recovery using the log procedure checkpoint facility 43
  Recovery without the checkpoint facility 43
Additional capabilities ....................... 43

Chapter 5. IMS performance feature log
and record definitions ...................... 45
Log definitions ................................ 45
Record definitions ............................. 46
  Comparison of performance programs 46
  Descriptions of record definitions ......... 47
Composite record definitions ................ 59
  Composite record sections in IMS_Vnnn_TRAN 59
  Composite record types and subtypes in IMS_Vnnn_TRAN 64

Chapter 6. IMS performance feature
data tables and lookup tables .......... 65
Naming standard for tables .................... 65
Table descriptions ............................ 65
Control tables ................................ 66
IMS log records component data tables .......... 66
IMS collect component data tables .......... 66
  Table and key column cross-reference 68
  Transaction subcomponent tables .......... 69
  IMS TRANSACTION_H_D_W .................. 69
  IMS_USER_TRAN_H_D_W ................... 80
  IMS_TRAN_TYPE key column ................ 81
System subcomponent tables .................. 83
  IMS_SYSTEM_Q_D 83
Application subcomponent tables .......... 85
  IMS APPLICATION_H_W .................... 85
Part 2. IMS Shared Queue feature

Chapter 7. Introduction to the IMS Shared Queue feature

Chapter 8. Using log and record procedures within the IMS Shared Queue

Chapter 9. Understanding data flow through IMS Shared Queue

Chapter 10. Administering the IMS Shared Queue feature

Part 3. Appendixes

Appendix A. Reports

Chapter 11. IMS Shared Queue record definitions

Chapter 12. IMS Shared Queue data tables and lookup tables
| IMS Application Response Time Trend report | IMS Message Queue Utilization Overview, Daily Report. | 171 | 204 |
| IMS Application Transaction Trend report | IMS Transaction Arrival Rate and Message Queue Usage, Daily Trend | 173 | 205 |
| IMS Application CPU Utilization Trend report | IMS CSQ Transaction Transit Time Reports | 174 | 206 |
| IMS System Response Time Trend report | IMS CSQ Transaction Transit Time Analysis By Transaction Name | 175 | 206 |
| IMS System Transaction Volumes Trend report | IMS CSQ Transaction Transit Time Analysis By LTERM and Userid | 177 | 209 |
| IMS System CPU Utilization Trend report | IMS CSQ Transaction Transit Time Analysis by Region | 179 | 213 |
| IMS System DLI Utilization Trend report | IMS CSQ Transaction Utilization Reports | 180 | 216 |
| IMS detail reports | IMS CSQ Resource Utilization, Daily Overview | 181 | 216 |
| IMS User ID Response Time and CPU Detail by Date report | | 181 | |
| IMS Transaction Utilization Detail by Date report | | 181 | |
| IMS Message Queue Pool Detail by Date report | | 184 | |
| IMS OSAM/ISAM Buffer Pool Detail by Date report | | 186 | |
| IMS VSAM Buffer Pool Detail by Date report | | 188 | |
| IMS Region Utilization Detail by Date report | | 190 | |
| IMS Region Detail by Date report | | 191 | |
| IMS Resource Detail by Quarter Hour report | | 192 | |
| IMS worst case reports | | 193 | |
| IMS User ID Resource Worst Case by Date report | | 193 | |
| IMS Program Utilization Worst Case by Date report | | 194 | |
| IMS Availability reports | | 195 | |
| IMS CSQ Subsystem Availability, Daily Trend Report | | 195 | |
| IMS CSQ Region Availability, Daily Overview report | | 197 | |
| IMS CSQ Application Usage and Availability report | | 198 | |
| IMS Message Queue Reports | | 200 | |
| IMS Message Queue Utilization, Date report | | 201 | |
| IMS Message Queue Utilization by Transaction, Date report | | 203 | |

Appendix B. Creating IMS log record DSECTs

Appendix C. DRLJXIDC DSECT macro

Appendix D. Sample archive exit

Appendix E. DFSLTMG0 log merge utility

Appendix F. List of abbreviations

Notices

Trademarks

Glossary

Index
### Figures

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overview of data collection using the IMS Performance feature</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Task flow for Tivoli Decision Support for OS/390 features</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>IMS_APPLICATION lookup table example</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>Example of log procedure parameter report</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Example of log procedure pending node report</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Example of log procedure composite record report</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>Example of R0 report</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>Example of R1 report</td>
<td>21</td>
</tr>
<tr>
<td>9</td>
<td>Overview of Tivoli Decision Support for OS/390 data flow</td>
<td>24</td>
</tr>
<tr>
<td>10</td>
<td>Data flow through the log collector</td>
<td>26</td>
</tr>
<tr>
<td>11</td>
<td>Data flow through DRL2LOGP</td>
<td>27</td>
</tr>
<tr>
<td>12</td>
<td>Data flow—statistics subcomponent</td>
<td>28</td>
</tr>
<tr>
<td>13</td>
<td>Data flow—application, transaction, and system subcomponents</td>
<td>29</td>
</tr>
<tr>
<td>14</td>
<td>Sample job for running the log collector</td>
<td>39</td>
</tr>
<tr>
<td>15</td>
<td>Sample data set with log collector parameters</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>Sample job for running DRL2LOGP</td>
<td>41</td>
</tr>
<tr>
<td>17</td>
<td>Key columns in transaction, system, and, application subcomponent tables</td>
<td>68</td>
</tr>
<tr>
<td>18</td>
<td>Overview of data collection using the IMS Shared Queue feature</td>
<td>104</td>
</tr>
<tr>
<td>19</td>
<td>Task flow for Tivoli Decision Support for OS/390 features</td>
<td>106</td>
</tr>
<tr>
<td>20</td>
<td>Example of log procedure parameter report for within IMS Shared Queue</td>
<td>114</td>
</tr>
<tr>
<td>21</td>
<td>Example of log procedure pending node report within IMS Shared Queue</td>
<td>114</td>
</tr>
<tr>
<td>22</td>
<td>IMS Shared Queue data flow through the log collector</td>
<td>116</td>
</tr>
<tr>
<td>23</td>
<td>Data flow through DRLSLOGP</td>
<td>117</td>
</tr>
<tr>
<td>24</td>
<td>Data flow—Account and Availability subcomponent</td>
<td>118</td>
</tr>
<tr>
<td>25</td>
<td>Data flow—Transaction Transit Time subcomponent</td>
<td>119</td>
</tr>
<tr>
<td>26</td>
<td>Sample job for running the log collector within IMS Shared Queue</td>
<td>125</td>
</tr>
<tr>
<td>27</td>
<td>Sample jobs for running DRLSLOGP</td>
<td>129</td>
</tr>
<tr>
<td>28</td>
<td>Tabular reports example</td>
<td>161</td>
</tr>
<tr>
<td>29</td>
<td>Graphic reports example</td>
<td>161</td>
</tr>
<tr>
<td>30</td>
<td>IMSY01 Report Query (DRLQM01Y01 member)</td>
<td>163</td>
</tr>
<tr>
<td>31</td>
<td>Example of Query from IMS_SYSTEM_TRAN_D Shared Queue Table</td>
<td>164</td>
</tr>
<tr>
<td>32</td>
<td>Example of Mixed Query</td>
<td>165</td>
</tr>
<tr>
<td>33</td>
<td>Example of Query Input Variables Panel</td>
<td>166</td>
</tr>
<tr>
<td>34</td>
<td>Example of IMS Application Response Time Overview graphic report</td>
<td>167</td>
</tr>
<tr>
<td>35</td>
<td>Example of IMS Application Transaction Overview graphic report</td>
<td>169</td>
</tr>
<tr>
<td>36</td>
<td>Example of IMS Application Response Time Trend graphic report</td>
<td>171</td>
</tr>
<tr>
<td>37</td>
<td>Example of IMS Application Transaction Trend graphic report</td>
<td>173</td>
</tr>
<tr>
<td>38</td>
<td>Example of IMS Application CPU Utilization Trend graphic report</td>
<td>174</td>
</tr>
<tr>
<td>39</td>
<td>Example of IMS System Response Time Trend graphic report</td>
<td>175</td>
</tr>
<tr>
<td>40</td>
<td>Example of IMS System Transaction Volumes Trend graphic report</td>
<td>177</td>
</tr>
<tr>
<td>41</td>
<td>Example of IMS System CPU Utilization Trend graphic report</td>
<td>179</td>
</tr>
<tr>
<td>42</td>
<td>Example of IMS System DLI Utilization Trend graphic report</td>
<td>180</td>
</tr>
<tr>
<td>43</td>
<td>Example of IMS User ID Response Time and CPU Detail by Date tabular report</td>
<td>181</td>
</tr>
<tr>
<td>44</td>
<td>Example of IMS Transaction Utilization Detail by Date tabular report</td>
<td>183</td>
</tr>
<tr>
<td>45</td>
<td>Example of IMS Message Queue Pool Detail by Date tabular report</td>
<td>184</td>
</tr>
<tr>
<td>46</td>
<td>Example of IMS OSAM/ISAM Buffer Pool by Date tabular report</td>
<td>186</td>
</tr>
<tr>
<td>47</td>
<td>Example of IMS VSAM Buffer Pool by Date tabular report</td>
<td>187</td>
</tr>
<tr>
<td>48</td>
<td>Example of IMS Region Utilization Detail by Date tabular report</td>
<td>190</td>
</tr>
<tr>
<td>49</td>
<td>Example of IMS Region Detail by Date tabular report</td>
<td>191</td>
</tr>
<tr>
<td>50</td>
<td>Example of IMS Resource Detail by Quarter Hour tabular report</td>
<td>192</td>
</tr>
<tr>
<td>51</td>
<td>Example of IMS User ID Resource Worst Case by Date tabular report</td>
<td>193</td>
</tr>
<tr>
<td>52</td>
<td>Example of IMS Program Utilization Worst Case by Date tabular report</td>
<td>194</td>
</tr>
<tr>
<td>53</td>
<td>Example of an IMS CSQ subsystem Availability, Daily Trend Report</td>
<td>195</td>
</tr>
<tr>
<td>54</td>
<td>Example of an IMS CSQ Region Availability, Daily Overview report</td>
<td>197</td>
</tr>
<tr>
<td>55</td>
<td>Example of an IMS CSQ application Usage and Availability report</td>
<td>198</td>
</tr>
<tr>
<td>56</td>
<td>Example of IMS Message Queue Utilization, Daily Report</td>
<td>201</td>
</tr>
<tr>
<td>57</td>
<td>Example of IMS Message Queue Utilization by Transaction, Date report</td>
<td>203</td>
</tr>
<tr>
<td>58</td>
<td>Example of an IMS Message Queue Utilization Overview, Daily Report</td>
<td>204</td>
</tr>
<tr>
<td>59</td>
<td>Example of an IMS Transaction Arrival Rate and Message Queue Usage, Daily Trend</td>
<td>205</td>
</tr>
<tr>
<td>60</td>
<td>Example of IMS CSQ Transit Time Analysis by Transaction Name, Daily report</td>
<td>206</td>
</tr>
<tr>
<td>61</td>
<td>Example of IMS CSQ Transit Time Analysis by LTERM and Userid, Daily report</td>
<td>210</td>
</tr>
<tr>
<td>62</td>
<td>Example of IMS CSQ Transit Time Analysis by Region, Daily report</td>
<td>213</td>
</tr>
<tr>
<td>63</td>
<td>Example of IMS CSQ Resource Utilization, Daily Overview</td>
<td>217</td>
</tr>
<tr>
<td>64</td>
<td>Sample JCL for assembling IMS log record DSECTs</td>
<td>220</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>65.</td>
<td>DRIJXIDC MACRO</td>
<td>. . . . . . . . . . . .</td>
</tr>
<tr>
<td>66.</td>
<td>Sample IMS archive exit</td>
<td>. . . . . . . . . . . .</td>
</tr>
</tbody>
</table>
## Tables

1. Logical sets for a full function transaction ........................................... 12
2. Main record set combinations ............................................................... 13
3. Parameter summary for DRL2LOGP ......................................................... 31
4. Parameter summary for the log procedure .............................................. 31
5. Comparison of the IMS Performance feature with other products for IMS record types ................................................................. 46
6. IMS record types and IMS Performance feature record definitions .......... 47
7. Composite record sections in IMS_Vnnn_TRAN .................................... 60
8. Composite record types and subtype sections in IMS_Vnnn_TRAN .......... 64
9. GROUP_ID lookup table ........................................................................ 99
10. Parameter summary for DRLSLOGP ....................................................... 122
11. Parameter summary for the log procedure within IMS Shared Queue .......... 122
12. Composite record sections ................................................................. 134
Preface

This book provides information about the IMS™ Performance and IMS Shared Queue features of IBM® Tivoli® Decision Support for OS/390® (hereafter referred to as Tivoli Decision Support for OS/390).

This book allows you to install and use the IMS Performance and IMS Shared Queue features of Tivoli Decision Support for OS/390. It describes:

- How to collect and report performance data generated by Information Management System/ESA (IMS/ESA®)
- Performance characteristics shown in Tivoli Decision Support for OS/390 reports, so you can analyze the characteristics of your system

Tivoli Decision Support for OS/390 was previously known as Tivoli Performance Reporter for OS/390.

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

Who should read this book

IMS Performance 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 OS/390 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 OS/390 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.

What this book contains

This book is organized in parts:

- Part 1, “IMS Performance feature” describes the Performance feature and contains the following chapters:
  - Chapter 1, “Introduction to the IMS Performance feature”
  - Chapter 2, “Using log and record procedures within the IMS performance feature”
  - Chapter 3, “Understanding data flow through IMS performance feature”
  - Chapter 4, “Administering the IMS Performance feature”
  - Chapter 5, “IMS performance feature log and record definitions”
  - Chapter 6, “IMS performance feature data tables and lookup tables”
Part 2, “IMS Shared Queue feature” describes the Shared Queue Feature and contains the following chapters:

- Chapter 7, “Introduction to the IMS Shared Queue feature”
- Chapter 8, “Using log and record procedures within the IMS Shared Queue”
- Chapter 9, “Understanding data flow through IMS Shared Queue”
- Chapter 10, “Administering the IMS Shared Queue feature”
- Chapter 11, “IMS Shared Queue record definitions”

Part III. Appendixes, contains the following appendixes:

- Appendix A, “Reports”
- Appendix B, “Creating IMS log record DSECTs”
- Appendix C, “DRLJXIDC DSECT macro”
- Appendix D, “Sample archive exit”
- Appendix E, “DFSLTMG0 log merge utility”
- Appendix F, “List of abbreviations”

The glossary and the index follow the appendixes.

Publications

This section lists publications in the Tivoli Decision Support for OS/390 library and any other related documents. It also describes how to access Tivoli publications online and how to order Tivoli publications.

Tivoli Decision Support for OS/390 library

The following documents are available in the Tivoli Decision Support for OS/390 library:

- Administration Guide, SH19-6816
  Provides information about initializing the Tivoli Decision Support for OS/390 database and customizing and administering Tivoli Decision Support for OS/390.
- Guide to the Reporting Dialog, 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.
- Language Guide and Reference, SH19-6817
  Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.
- User’s Guide for the Viewer, SH19-4517
  Provides information about how use the graphical interface for Tivoli Decision Support for OS/390.
- Messages and Problem Determination, SH19-6902
  Provides information to help operators and system programmers understand, interpret, and respond to Tivoli Decision Support for OS/390 messages and codes.
- Accounting Feature for the Host, SH19-4495
  Provides information for users who want to use Tivoli Decision Support for OS/390 to collect and report performance data generated by the Accounting feature.
- Accounting Feature for the Workstation, SH19-4516
Provides information for users who want to use the Accounting Workstation Option to manage, process, and analyze financial data on a workstation.

- **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.

- **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).

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

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

- **Network Performance Feature Guide, 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 Reference Vol. I, SH19-6819**
  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for OS/390 to analyze Multiple Virtual Storage (MVS) or Virtual Machine (VM) performance data.

- **System Performance Feature Reference Vol. II, SH19-4494**
  Provides information for administrators and users with a variety of backgrounds who want to use Tivoli Decision Support for OS/390 to analyze Multiple Virtual Storage (MVS) or Virtual Machine (VM) performance data.

- **IBM Online Ominbus Edition z/OS Collection Kit, SK2T-6700**
  CD containing all z/OS™ documentation.

- **Networking Systems Collection Kit, SK2T-6012**
  CD containing all networking systems documentation.

### Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.
You can access LookAt from the Internet at:

http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/ or from anywhere in z/OS or z/OS.e where you can access a TSO/E command line (for example, TSO/E prompt, ISPF, z/OS UNIX® System Services running OMVS).

The LookAt Web site also features a mobile edition of LookAt for devices such as Pocket PCs, Palm OS, or Linux-based handhelds. So, if you have a handheld device with wireless access and an Internet browser, you can now access LookAt message information from almost anywhere.

### 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. The Tivoli Software Information Center is located at the following Web address:

http://publib.boulder.ibm.com/tividd/td/tdprodlist.html

Click the Tivoli Decision Support for OS/390 link to access the product library.

These publications are available in PDF or HTML format, or both. Translated documents are also available for some products.

**Note:** If you print PDF documents on other than letter-sized paper, select the *Fit to page* check box in the *Adobe Acrobat Print* dialog. This option is available when you click *File* → *Print*. *Fit to page* ensures that the full dimensions of a letter-sized page print on the paper that you are using.

### Ordering publications

You can order many Tivoli publications online at the following Web site:


You can also order by telephone by calling one of these numbers:

- In the United States: 800-879-2755
- In Canada: 800-426-4968

In other countries, see the following Web site for a list of telephone numbers:

http://www.ibm.com/software/tivoli/order-lit/

### 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 *Administration Guide*.
Contacting software support

If you have a problem with any Tivoli product, refer to the following IBM Software Support Web site:


If you want to contact software support, see the IBM Software Support Guide at the following Web site:


The guide provides information about how to contact IBM Software Support, depending on the severity of your problem, and the following information:

- Registration and eligibility
- Telephone numbers and e-mail addresses, depending on the country in which you are located
- Information you must have before contacting IBM Software Support

Note: For Tivoli NetView® for OS/390 customers only, additional support is available on the NETVIEW CFORUM (Customer Forum) through the IBMLink™ system. This forum is monitored by NetView developers who answer questions and provide guidance. When a problem with the code is found, you are asked to open an official problem management record (PMR) to obtain resolution.

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 term z/OS is used in this book to mean z/OS and OS/390 operating systems. Where the term OS/390 does appear, the related information applies only to OS/390 operating systems.

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
Changes in this edition

This edition is an updated version that replaces the previous edition of the same book. The changes are:

- New information about the IMS Shared Queue and non-shared queue configuration, such as IMS Availability, IMS Light Feature, IMS V8 support, IMS Log merging and IMS queues utilization, in Chapter 7, “Introduction to the IMS Shared Queue feature”, Chapter 8, “Using log and record procedures within the IMS Shared Queue” and Chapter 9, “Understanding data flow through IMS Shared Queue”, on page 115.

- New data tables and lookup tables about the Shared Queue component in Chapter 12, “IMS Shared Queue data tables and lookup tables”.

- The sections about the Availability Reports, the Message Queue reports, the CSQ Transaction Transit Time Reports and the CSQ Utilization Reports in Appendix A, “Reports”.

The name of the base product has been changed to Tivoli Decision Support for OS/390.

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.
Introductory paragraph here
Chapter 1. Introduction to the IMS Performance feature

IBM Tivoli Decision Support for OS/390 (hereafter referred to as Tivoli Decision Support for OS/390) 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 OS/390 consists of a base product and several optional features.

The Tivoli Decision Support for OS/390 base includes:
• Interactive System Productivity Facility (ISPF) host reporting and administration dialogs
• The Tivoli Decision Support for OS/390 log collector program
• Log and record definitions for all records used by the Tivoli Decision Support for OS/390 features

Each feature provides:
• Update definitions for DB2® tables
• Table definitions
• Report definitions

Tivoli Decision Support for OS/390 enables you to collect large volumes of data and keep the space to store it at acceptable levels. This Tivoli Decision Support for OS/390 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 OS/390 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.

Understanding the IMS Performance feature

The IMS Performance feature collects IMS performance data to produce reports. Reports are produced using information stored in Tivoli Decision Support for OS/390 DB2 tables. Figure 1 on page 4 shows an overview of the IMS Performance feature.
Collecting data

The process of collecting IMS performance data into DB2 tables is called a *collect*. It works like this:

**SLDS**
IMS produces a system log data set (SLDS) during the IMS archive process. The IMS Performance feature uses the IMS SLDS as input.

**Log procedure**
A Tivoli Decision Support for OS/390 log processing program called a log procedure processes selected records in the IMS SLDS. The log procedure matches records that have been written for IMS events. An IMS event is an activity that is part of a transaction or an IMS system activity. Each IMS record type represents an IMS event. Together, a number of records contain all information about a transaction.

The log procedure saves the matched records until the transaction is complete.
Composite record
The log procedure creates a composite record from these matched records in the SLDS log—when the transaction is complete.

Record procedure
A Tivoli Decision Support for OS/390 record processing program called a record procedure processes the composite records and creates simplified records called R2 records.

Record definitions
The IMS Performance feature provides record definitions for the individual record types found in the IMS SLDS, and for the additional records created by the Tivoli Decision Support for OS/390 record procedure. The record definitions are used by Tivoli Decision Support for OS/390 collect when updating DB2 tables.

Tivoli Decision Support for OS/390 data tables and environmental information
The IMS Performance feature uses the R2 records, along with user-supplied data in Tivoli Decision Support for OS/390 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.

Reports
Tivoli Decision Support for OS/390 creates reports from the information in the data tables. In addition to the reports provided with the IMS Performance feature, you can create your own reports using, for example, the Query Management Facility (QMF™) prompted query language.

The log collector and DRL2LOGP
To collect data as described in Figure 1 on page 4 you run the Tivoli Decision Support for OS/390 log collector program. The log collector uses record definitions and other definitions when it updates Tivoli Decision Support for OS/390 data tables. The log collector is part of the Tivoli Decision Support for OS/390 base product.

An alternative way to process the IMS SLDS without using the log collector is to run a batch program provided with the IMS Performance feature, called DRL2LOGP.

DRL2LOGP is a stand-alone batch program that calls the log procedure and record procedures. DRL2LOGP does not update the DB2 tables. It produces output composite records. It also produces a statistics report and a detailed transaction report.

DRL2LOGP is usually used only for detailed analysis because it produces a large amount of output. You do not have to install the IMS Performance feature or its components to use DRL2LOGP.

For data flow diagrams for the log collector and DRL2LOGP, see Chapter 3, “Understanding data flow through IMS performance feature”, on page 23. For more information about running the log collector and DRL2LOGP, see Chapter 4, “Administering the IMS Performance feature”, on page 31.
Installing and customizing the IMS Performance feature

This section supplements the general feature installation procedure described in the Administration Guide for installing and customizing a Tivoli Decision Support for OS/390 feature component.

To install and use the IMS Performance feature, you must have an MVS operating system capable of running Tivoli Decision Support for OS/390. The IMS Performance feature supports data from systems running IMS/ESA from Version 5 Release 1 to Version 7 Release 1. You use the IMS system log data set (SLDS) to generate data for the predefined tables and reports in the IMS Performance feature.

Figure 2 shows the sequence of events in planning for, installing, customizing, and administering a Tivoli Decision Support for OS/390 feature.

Planning for the IMS Performance feature

Your most critical planning task is determining what kind of information users need from the IMS Performance feature. For example, users may be interested only in system resource availability or transaction response time. Installing only those parts of the feature needed to meet user requirements ensures that the feature benefits users while minimizing the performance impact caused by data collection and interpretation activities.

After you have installed the IMS Performance feature using SMP/E, plan each step of the implementation process:

1. Determine what users need from the IMS Performance feature. What tasks must they perform that the feature can accomplish or assist with?

2. Determine what components and subcomponents you must install to meet users’ needs. See “Selecting IMS Performance feature components” on page 7 for a description of the components and subcomponents available.
3. Determine the administration tasks you must perform to customize Tivoli Decision Support for OS/390 and the IMS Performance feature to work with your computer system. Make any decisions necessary to perform these tasks.

4. For the selected components, determine the customization tasks required to customize the supported products to work with Tivoli Decision Support for OS/390 and with the IMS Performance feature.

If you are planning for the first time, you must perform all these steps to ensure that your implementation of the feature is consistent and is driven by a common goal. If you are reading this chapter in preparation for modifying your system, you may not need to perform all of these tasks.

The detailed planning tasks you must perform depend on the components you choose to install. However, the basic planning process is the same for all components.

When you are ready to install and customize an IMS Performance feature component, refer to the procedures in the Administration Guide.

Selecting IMS Performance feature components

The IMS Performance feature is divided into components and subcomponents. Consider carefully which of these to install. If you need reports from a component that you have not installed, you must install the component and then wait several days or weeks until enough data has been collected to create reports. Alternatively, if you install more components than you need, Tivoli Decision Support for OS/390 collects unnecessary data, which takes up disk space.

The IMS Performance feature components and subcomponents contain Tivoli Decision Support for OS/390 objects (for example, predefined reports, tables, and update definitions). Each IMS Performance feature component contains the objects required to collect performance and service level data from the appropriate records in the IMS log and produce reports.

You can install all Tivoli Decision Support for OS/390 features and components using the procedure in Administration Guide. After the system programmer has successfully installed the Tivoli Decision Support for OS/390 base, you can choose whether to install the IMS feature and its components and subcomponents. Tivoli Decision Support for OS/390 stores the necessary log, record, and update definitions in Tivoli Decision Support for OS/390 system tables. Tivoli Decision Support for OS/390 also loads predefined DB2 tables and reports.

The IMS Performance feature components are:
- IMS 5.1 collect component
- IMS 5.1 log records component
- IMS 6.1 collect component
- IMS 6.1 log records component
- IMS 7.1 collect component
- IMS 7.1 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 subcomponent Collects information about transactions and BMPs. Information available includes system response
Introduction to the IMS Performance Feature

times, system transaction volumes, CPU and
database utilization, and transaction detail.

System subcomponent Collects information about general system activity.
Information available includes system response
times and region utilization.

Application subcomponent Collects information about IMS application
programs, including response time, transactions,
and CPU utilization.

Statistics subcomponent Records statistical information about buffer and
pool usage.

The log records components
You can use the log records components for your own IMS analysis. When you
install a log records component, you get access to the Tivoli Decision Support for
OS/390 record definitions for IMS records in the SLDS. You can write your own
Tivoli Decision Support for OS/390 definitions to process IMS records, and, for
example, define your own DB2 tables and reports.

See Chapter 5, "IMS performance feature log and record definitions", on page 45
for descriptions of the Tivoli Decision Support for OS/390 record definitions for
the IMS records.

The log records components have no subcomponents. They do not update DB2
tables or produce reports.

Updating lookup tables
To accurately analyze performance data from your system, you should group data
by user groups, workload types, project groups, and so on. Each installation has
different criteria for grouping data. The application subcomponent of the IMS
Performance feature includes a lookup table (IMS_APPLICATION) that you can
customize to specify the groupings you want reflected in your reports.

To decide how you want data grouped, develop a performance and service level
strategy. For general information about developing a performance and service level
strategy, refer to the System Performance Feature Guide. After developing your
strategy, modify the lookup table to carry out your strategy. Lookup table
modification is the only customization that you must perform for the IMS
Performance feature.

As the needs of your organization change, so will your service level strategy. You
may need to update the Tivoli Decision Support for OS/390 lookup tables
periodically to reflect these changes.

Updating the IMS_APPLICATION lookup table
The IMS_APPLICATION lookup table is used only for the application
subcomponent. It groups transactions by application, and can also group them into
subsets by program name. The administration dialog prompts you to edit this
lookup table when you install the IMS Performance feature application
subcomponent online. Refer to the Administration Guide for more information about
editing lookup tables. Figure 3 on page 9 shows an example of a completed
IMS_APPLICATION lookup table.
Updating other lookup and control tables
The IMS Performance feature uses the DAY_OF_WEEK and PERIOD_PLAN control tables, which are also used by other Tivoli Decision Support for OS/390 features. Check these tables and update them as needed.

For information about these tables, refer to the Administration Guide.

If you have installed the Tivoli Decision Support for OS/390 System Performance feature, you can use it to collect and report on data regarding 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 about the table, refer to the System Performance Feature Reference Volume I.

Using the IMS Performance feature
Before starting the daily use of the IMS Performance feature, run a few tests to ensure that the installation was successful. Verify that Tivoli Decision Support for OS/390 is collecting the right data, storing the data correctly, and using the proper data to generate the reports. Verify also that the lookup table contains the appropriate groups.

After you verify that the installation was successful, you can put the IMS Performance feature into production.

Refer to the Administration Guide for the steps in testing component installation and for general instructions for running Tivoli Decision Support for OS/390. For specific information about running the IMS Performance feature, see.
Chapter 2. Using log and record procedures within the IMS performance feature

This chapter explains the use of log procedures and record procedures within the IMS Performance feature. A log procedure takes two or more records from a log and creates one 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 DRL3I512, DRL3I612, and DRL3I712 record procedures take composite records created by the log procedure and simplify them to make collection and reporting easier. You can also add record procedures that can be used for different purposes.

The log procedure

The IMS Performance feature is based on a log processing routine (log procedure) designed to process selected records on the IMS SLDS. The procedure produces composite records at IMS transaction level (full function or Fast Path), rather than at the program specification block (PSB) level, and therefore the records are more detailed and meaningful. The log procedure copies most relevant IMS log records in their entirety to the composite record. The X'01', X'03', X'13', X'5901', and X'5903' records are exceptions. The log procedure copies the text prefix (but not the text) from these records to the composite record, including all headers.

Record grouping

The principal IMS records matched by the IMS log procedure are classified under logical sets. These logical sets are a collection of related records that represent database and data communication activity taking place for an IMS transaction. The classification is based on the function of the records, and is required for simplification of the matching process. Each logical set has a unique key, and the records that fall in a logical set carry the same unique data value or key. These are the logical sets:

Input set (D1)
Consists of X'01'/X'03' destined for a scheduler message block (SMB or transaction), X'35' In and X'31' In, which all contain the same disk-relative record number (DRRN) used as the key. This set represents the arrival (X'01' or X'03') of an input message, its queuing (X'35' In) in MSGQ, and the GU of the message by DL/I to dequeue it from the input queue (X'31' In) and give it to the PSB for processing.

For message switches, where the input message is destined for a communications name table (CNT) rather than a transaction, the type X'31' Out record represents the GU of the message by IMS to get the message from the output queue. Here, a type X'36' record (indicating dequeue of the message on a CNT) also appears in the input set.

Output set (D2)
Consists of X'03', X'35' Out, X'31' Out, and X'36' Out.

These records all carry the same DRRN, which is used as the key. The records represent the generation (X'03') of an output message, its enqueuing (X'35' Out), the GU by IMS to retrieve it from the output queue (X'31' Out), and dequeuing (X'36' Out) on a CNT in MSGQ.
For program switches, where X’03’ is destined for an SMB (transaction) instead of a CNT, X’31’ represents the GU of the message by IMS DL/I to get it from the input queue and give it to the PSB for processing. Here, the X’36’ (indicating dequeue of the message on CNT) is absent.

**PSB set (RTKN)**
Consists of X’08’ and X’07’. Both records carry the same high order 12-byte recovery token, which is used as the key. These records represent the scheduling (X’08’) and termination of a PSB (X’07’) in the IMS DB system.

**EMH set (EMH)**
Consists of X’5901’, X’5903’, and X’5936’. These records carry the same full 16-byte recovery token, which is used as the key. It represents the arrival of the input message (X’5901’), the generation of the output message (X’5903’), and the dequeuing of the output message (X’5936’) on the EMH.

**Conversation set**
Consists of X’11’, X’12’, and X’13’. The X’11’ and X’12’ carry the same input node name and offset to the conversational control block (CCB). The X’13’ is matched to the transaction using the CCB ID. The CCB ID is calculated from the CCB offset found in the X’11’ (conversation start) record.

**Unit-of-recovery (UOR) set**
Consists of X’37XX’, X’5937’, and X’56’. All of these records carry the same recovery token of a full 16 bytes, which is used as the key. The UOR set represents activity relating to a PSB set since the last commit point. It includes commit records (X’37’ for full function, X’5937’ for Fast Path) or failure records (X’38’ for full function, X’5938’ for Fast Path) for DB/DC activity incurred since the last commit point of a given scheduled PSB in IMS. The external subsystem (X’56’) records written out since the last commit point also belong to this set. The relation between the PSB set and the UOR set is 1:n. Thus, this set is dependent on the presence of the PSB set.

The input set, output set, and PSB set are stand-alone sets for full function transactions and the EMH set is a stand-alone set for Fast Path transactions. Because they are stand-alone sets, they are the main matching sets for any IMS transaction. Table 1 shows how the logical sets are represented in a simple full function transaction that issues output messages.

**Table 1. Logical sets for a full function transaction**

<table>
<thead>
<tr>
<th>Event</th>
<th>Input set (D1)</th>
<th>PSB set (RTKN)</th>
<th>Output set (D2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message received by IMS</td>
<td>X’01’/X’03’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enqueue input message</td>
<td>X’35’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schedule application message</td>
<td></td>
<td>X’08’</td>
<td></td>
</tr>
<tr>
<td>Retrieve input message using DL/I call</td>
<td>X’31’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message received from application</td>
<td></td>
<td></td>
<td>X’03’</td>
</tr>
<tr>
<td>Enqueue output message</td>
<td></td>
<td></td>
<td>X’35’</td>
</tr>
<tr>
<td>Application terminates</td>
<td>X’07’</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMS issues GU for message</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complete message sent to destination</td>
<td></td>
<td></td>
<td>X’36’</td>
</tr>
</tbody>
</table>
All IMS full function transactions, message switches, and BMP programs basically consist of this same type of pattern, with some exceptions and variations.

Set relationships

Certain relationships exist between the different logical sets of record groups, depending on the type of transaction:

Input and output sets (D1-D2)
The link between input and output sets is valid only for a full function transaction. When possible, this link is determined indirectly through the presence of the same 16-byte recovery token on a X'31' In record (from the input set) and a X'35' Out record (from the output set). The X'01' (from input set) and X'03' (from output set) carry the same input node name/input node sequence number and MSGPREFI values when the output set is related to the input set, and this relationship may also determine the link.

Input and PSB sets (D1-RTKN)
This relationship is valid only for full function transactions where the input message is destined for a transaction (SMB). The X'31' In of the input set serves to create the link, as it carries a 16-byte recovery token whose high-order 12 bytes are also the PSB set key. Note that if a X'08' is missed (normally because it is in an earlier log), the X'31' causes the PSB entry to be created.

Output and PSB sets (D2-RTKN)
This relationship is valid only for full function transactions and BMP programs that produce output messages. The X'35' Out of the output set serves to create the link, as it carries the 16-byte recovery token whose high-order 12 bytes are also the PSB set key.

EMH and PSB sets (EMH-RTKN)
This relationship is always valid. The high-order 12 bytes of the EMH set key form the PSB set key.

EMH and output sets (EMH-D2)
This relationship is only valid for EMH transactions that create output messages using MSGQ facility. The X'35' Out of the output set carries the 16-byte recovery token that is the EMH set key.

Composite records and subtypes

The main sets of records (input set, output set, EMH set and PSB set) can combine in a number of ways. The log procedure tries to match records to these combinations and write composite records appropriately, as shown in Table 2.

<table>
<thead>
<tr>
<th>Composite record subtype</th>
<th>Record set combination</th>
<th>Typical description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'EC'</td>
<td>EMH only</td>
<td>EMH transactions where X'08' was missed</td>
</tr>
<tr>
<td>X'ED'</td>
<td>EMH and D2</td>
<td>EMH transactions that produce outputs, but where X'08' was missed</td>
</tr>
<tr>
<td>X'EE'</td>
<td>EMH and RTKN</td>
<td>EMH transactions</td>
</tr>
<tr>
<td>X'EF'</td>
<td>EMH, RTKN, and D2</td>
<td>EMH transactions that produce outputs</td>
</tr>
<tr>
<td>X'FA'</td>
<td>D1 only</td>
<td>Message switch or MSC message switch</td>
</tr>
</tbody>
</table>
Table 2. Main record set combinations (continued)

<table>
<thead>
<tr>
<th>Composite record subtype</th>
<th>Record set combination</th>
<th>Typical description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'FB'</td>
<td>D2 only</td>
<td>MTO and AOI message switches</td>
</tr>
<tr>
<td>X'FC'</td>
<td>RTKN only</td>
<td>Non-message-driven BMP programs that produce no output messages</td>
</tr>
<tr>
<td>X'FD'</td>
<td>D2 and RTKN only</td>
<td>Non-message-driven BMP programs that produce output messages</td>
</tr>
<tr>
<td>X'FE'</td>
<td>D1 and RTKN only</td>
<td>Transactions and message-driven BMP programs that produce no output messages</td>
</tr>
<tr>
<td>X'FF'</td>
<td>D1, D2, and RTKN</td>
<td>Transactions and message-driven BMP programs that produce outputs</td>
</tr>
</tbody>
</table>

Conversation set records are present only where both D1 and RTKN sets exist. Thus, conversation set records are present only for record subtypes X'FE' and X'FF'.

UOR set records are present in all subtypes where a PSB is scheduled for the IMS event (subtypes X'EC', X'ED', X'EE', X'EF', X'FC', X'FD', X'FE', and X'FF'). Record subtypes X'EC' and X'ED' are included here because the PSB is scheduled even though the schedule is missed (X'08' records).

UORs that cannot be matched with a transaction occur for non-BMP programs. These UORs have these composite record types:

- X'0C': Neither input nor output has occurred
- X'0D': No input has occurred, but there was output

Another type of composite record is created using X'45'/X'47' and X'4001' checkpoint records. The X'45' and X'47' records contain statistics about the IMS system; therefore, they do not need to be matched to other records to create a composite record. However, certain information from the accompanying X'4001' checkpoint record is included in these records, resulting in composite statistics records that have X'45' or X'47' data with 14 bytes of checkpoint information inserted after the record subtype.

Incomplete transactions have this composite record type:

- X'0F': Transaction was incomplete

The log procedure may write incomplete transactions because:

- Limits specified by the TABLEFLUSH or WRITEPENDING parameters have been reached
- The output length exceeded the limit
- Excess outputs forced unmatched transactions

Handling of special IMS cases

The log procedure handles special IMS cases as described here.

Multiple segment input

The first or only segment creates an Input-DRRN table entry. The log procedure matches subsequent segments on the message DRRN and chains...
them on the IMS record chain. It then combines multi-segment X'01' and X'03' records in such a way so that the number of segments and the total size of the text is available.

**Multiple segment output**
The first or only segment creates an Output-DRRN table entry. The log procedure matches subsequent segments on the message DRRN and chains them on the IMS record chain. It then combines multi-segment X'01' and X'03' records in such a way so that the number of segments and the total size of the text is available.

**Multiple outputs**
Each output creates an Output-DRRN table entry. Multiple outputs are valid only when a corresponding input set/PSB set is present. The multiple outputs are linked through pointers, and each new output is placed at the end of the current list. When the log procedure writes a composite record, it also searches the linked list and writes all output sets for a given input set, EMH set, or PSB set. Multiple linked outputs are possible only when outputs are from a scheduled PSB (PSB set is present). But because the procedure attempts to produce composites at transaction level rather than at PSB level, it links the outputs to an input message set (D1-DRRN) or EMH set. Outputs are only linked to a PSB when these sets are absent.

**Multiple transactions per schedule of a PSB**
The log procedure creates the RTKN entry upon receiving a X'08' record. The X'31' In record, which carries the input set DRRN and the recovery token of the PSB set, creates the linkage between an input-DRRN table entry and the corresponding RTKN entry through the UOR node. Writing of the composite record for a completed transaction is deferred until the PSB terminates (X'07' record is encountered), so that the PSB set is complete when written. (The extent to which the log procedure holds otherwise complete transactions can be controlled by the TABLEFLUSH parameter. See "Specifying DRL2LOGP and log procedure parameters" on page 31 for a detailed discussion of the TABLEFLUSH parameter.) See also the discussion of wait for input (WFI) programs on page 17.

**Program-to-program switch**
Distinctions are made between transactions that started with a X'01' record (root transaction) and those that started with a program switch (child transaction), as indicated by a X'03' record that has the MSGQDES flag set to X'81' (destination is an SMB) rather than X'82' (destination is a CNT). Although the log procedure writes the root transaction composite record when it is considered complete, it retains all the input set details until all child transactions created by the root have also been written as composite records. For example, if A is a root transaction that creates transaction B, and B creates transaction C, C is also treated as a child transaction created from A. The log procedure retains the root transaction’s input set details, because it writes the input set of the root transaction for the child transaction as well as its own input set.

Note that no summarizing occurs and that all transaction entities pass as composite records. The retained root transaction’s input set details are also not subject to the TABLEFLUSH parameter.

The input-DRRN table contains a list of pending child transactions. For the child transaction, the input-DRRN table entry points to the root transaction. For a program switch, the log procedure creates an input-DRRN entry using data from an output-DRRN table entry. (The log procedure treats the X'03' record like a X'01' record, and the X'35' Out
record like a X'35' In record. It then matches the subsequent X'31' record to
the input-DRRN entry, and also attempts to match the output-DRRN entry
if it has not yet been freed.)

AOI user-exit initiated transactions
The log procedure treats the X'03' record corresponding to the AOI user
exit like a X'01' record and builds an input-DRRN table entry.

Input message reenqueue
The X'38' record, which indicates a message reenqueue, creates a new
input-DRRN table entry that a subsequent X'31' In record will match.

Output message reenqueue
Here, (for example, when IMS finds that the terminal does not
acknowledge successful receipt of a message) IMS may:
1. Save the output message (indicated by a X'36' save record)
2. Reenqueue the same message to the same destination (indicated by the
appearance of a second X'35' record with the reenqueue flag set)
3. Get unique the message from the output queue again (indicated by the
appearance of a second X'31' record)
4. Dequeue the message, if the terminal acknowledges successful receipt
of the message (indicated by the presence of a second X'36' record)
5. Delete the message from the queue (indicated by the appearance of a
X'33' 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 GU;
thus the delay would be attributed to the network.

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

Non-message-driven BMP programs
The X'08' record creates the PSB (RTKN) entry, and the output X'03' creates
output-DRRN entries for the BMP program. These two entries are linked
using the recovery token on X'35' Out.

System-generated output (including master terminal operator (MTO) traffic)
The X'03' record creates output-DRRN entries. When the log procedure
receives the X'36' (DEQ) or X'33' (FREE), it writes the output-DRRN entry
as a composite record with subtype X'FB'.

Terminal message switch
The X'01' creates the input-DRRN entries. When the log procedure
receives the X'33' (FREE) for the input DRRN, it writes the input-DRRN entry
as a composite record with subtype X'FA'. This special case may also include
MSC and ISC message switching.

Conversational transactions
The log procedure creates a scratchpad area (SPA) entry on the arrival of a
X'11' record. The X'12' record triggers the completion of the SPA entry. The
log procedure then matches a full function transaction using the input
node from X'01' to a SPA entry, but only when the X'31' In record has the
flag indicating that the CCB ID is present. If the conversation is not
terminated by the transaction (a X'13' record is written instead of a X'12'),
the X'13' record is matched to the input-DRRN entry indirectly through the
RTKN entry. (First, the log procedure matches X'13' record to the RTKN
node using the high order 12-byte recovery token, and then to the UOR
node on the RTKN_UOR list using the last 4 bytes of the full 16-byte
recovery token. The log procedure obtains the SPA entry from the D1 node
pointed to by the matched UOR entry.) The composite record contains the
SPA section with data from X'11' and X'13'/X'12'.

Conversational transactions with program-to-program switch
No special treatment occurs. The X'13' record is missing, with SPA data
present on the X'03' record, which indicates the program-to-program
switch.

Fast Path (EMHs)
The X'5901' record creates an EMH entry (using the full 16-byte recovery
token) and the secondary index entry in table EMH_INODE. The log
procedure matches subsequent X'5903' record and X'5936' record to EMH
entry using the recovery token, if present. If the recovery token is absent,
the log procedure uses the EMH_INODE table to obtain the EMH entry.

MSGQ output produced by EMH transactions (if any) links through the
16-byte recovery token carried by X'35' Out record of output DRRN.

Wait-for-input (WFI) programs
If the program is a WFI, the log procedure does not hold incomplete
transactions until X'07' is encountered. Instead, the log procedure writes
the composite record when all outputs for the corresponding input are
completed.

If the log procedure cannot determine that the program is a WFI, it
assumes multiple transactions for a single PSB schedule. It holds
incomplete transactions until a X'07' is encountered or the TABLEFLUSH
parameter causes the pending transactions to be flushed. See “Specifying
DRL2LOGP and log procedure parameters” on page 31 for a detailed
discussion of the TABLEFLUSH parameter.

Quick reschedule
The X'07' indicates a case of quick reschedule. Because IMS may not write
the subsequent X'08', the log procedure creates the RTKN entry upon
receiving the X'07' record. The X'08' record, if it follows, is matched to the
already-created RTKN entry.

ISC and front-end switching (FES)
The IMS records written here 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 16.

Multiple Systems Coupling (MSC)
The IMS records written in the originating system are the same as for the
terminal message switch case. The log procedure treats the input message
and the reply from the remote system as seen in the originating system
exactly like a terminal message switch. For more information about
terminal message switch, see page 16 In the remote system, the sequence
of records written out is the same as for a full function transaction
sequence.

Mode multiple
Here, the log procedure detects the repeated use of the same recovery
token (UOR) for different transactions and creates an entry for each input
message and UOR. This allows each transaction to be uniquely identified
and reported, regardless of whether the UOR is unique.
Using log and record procedures within IMS

Release dependency

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 per the IMS release specified at run time. However, if IMS release changes invalidate the matching logic, an update or new release of the IMS Performance 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 several useful reports, and informational, warning, and error messages. For information about messages and codes issued by the log procedure, refer to the Messages and Problem Determination.

The log procedure parameter report (Figure 4) 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.

DRL2070I Batch Driver Parameters:

```
DRLIPARM Reports requested are  R0(00-FE)
DRLIPARM Reports requested are  R1(FF)
DRLIPARM Reports requested are  R2(FF)
DRLIPARM IMS version :  51
```

DRL2071I Parameters used in this run:

```
Default TABLE FLUSH   =  0
Default MAX UOR       =  5
Default MAX OUTPUT    =  5
Default START         =  0000000F  0000000F
Default STOP          =  0099365F  2359599F
DRLIPARM IMSID          =  IMSP
Default RECTYPE        =  FF
DRLIPARM IMSIDCHECK    =  CONTINUE
DRLIPARM WRITEPENDING  =  YES
Default PASSLOGREC     =  YES
Default FULLFUNC ESS SUPPRESS  =  NO
Default FULLFUNC MSGSWITCH SUPPRESS  =  NO
Default FULLFUNC BMP SUPPRESS  =  NO
Default FULLFUNC FPSYNC SUPPRESS  =  NO
Default FULLFUNC CONV SUPPRESS  =  NO
Default FULLFUNC BMP SUPPRESS  =  NO
Default FASTPATH MSGQ SUPPRESS  =  NO
Default FASTPATH FPSYNC SUPPRESS  =  NO
DRLIPARM MTO TRAFFIC SUPPRESS  =  NO
```

DRL2054I Processing log IMS.SLDS.TESTLOG on volume VOL001.
DRL2064I IMS System IMSP started at 0093242F 09514236 has switched OLDS at 0093242F 09520269 as indicated by type 42 record at 00000001.

Figure 4. Example of log procedure parameter report

The log procedure pending node report (Figure 5 on page 19) appears after the log procedure has completed and indicates the number of nodes pending in storage tables by type. Nodes are the internal representations of the data before they are grouped as a complete composite record. The log procedure writes these pending nodes to the checkpoint file allocated to DRLICHKO, (if present) which can be used when processing the next SLDS for the same IMS system.
The log procedure composite record report (Figure 6) also appears after the log procedure has completed and indicates the number of composite records that the log procedure created this run by composite record subtype. If you ran the log collector, this report indicates the amount of storage used for the composite records before the DB2 tables were updated. If you ran DRL2LOGP, this report indicates the number of records written.

Record procedures

Record procedures (also known to DRL2LOGP as report procedures) simplify the composite records created by the log procedure. They extract certain information from the composite records and original IMS records, passing the results back as output to be printed or further processed.

You can write your own record procedures, using the ILOGREC log records mapping macro and other macros supplied with IMS (see Appendix B, “Creating IMS log record DSECTs”, on page 219), and the DRLJXIDC macro supplied with the IMS Performance feature (see Appendix C, “DRLJXIDC DSECT macro”, on page 223).

The record procedures provided with the IMS Performance feature are described below.
The characters *nn* in the record procedure name indicate the IMS release number. *nn* can be 51, 61, or 71 where, for example, 51 signifies IMS Version 5 Release 1.

The names in parentheses, for example (R0), are short forms of the record procedure name that are used in this book and in reports.

**DRL3Inn0 (R0)**

The output ddname for this record procedure is DRLIRPT0, when run under DRL2LOGP control. It produces a statistical report containing run data, such as the number of records read, and the number of records read per record type. See Figure 7.

<table>
<thead>
<tr>
<th>IMS log records read by report R0</th>
<th>Min</th>
<th>Max</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 Message from a CNT</td>
<td>222</td>
<td>207</td>
<td>2682</td>
</tr>
<tr>
<td>02 Command</td>
<td>1</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>03 Message from an SMB or IMS</td>
<td>512</td>
<td>168</td>
<td>2907</td>
</tr>
<tr>
<td>06 IMS/ESA accounting</td>
<td>1</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>07 Application program end</td>
<td>163</td>
<td>248</td>
<td>40424</td>
</tr>
<tr>
<td>08 Application program schedule</td>
<td>162</td>
<td>88</td>
<td>14256</td>
</tr>
<tr>
<td>20 Data base open</td>
<td>8</td>
<td>54</td>
<td>432</td>
</tr>
<tr>
<td>31 Message queue Get Unique</td>
<td>668</td>
<td>48</td>
<td>61224</td>
</tr>
<tr>
<td>33 Message queue Free</td>
<td>666</td>
<td>20</td>
<td>34</td>
</tr>
<tr>
<td>35 Message queue Enqueue/Re-enqueue</td>
<td>720</td>
<td>60</td>
<td>52414</td>
</tr>
<tr>
<td>36 Message queue Dequeue/Save/Delet</td>
<td>464</td>
<td>84</td>
<td>40260</td>
</tr>
<tr>
<td>37 Message transferred</td>
<td>416</td>
<td>44</td>
<td>21092</td>
</tr>
<tr>
<td>00 Message XFER record</td>
<td>15</td>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>01 Message XFER record</td>
<td>108</td>
<td>56</td>
<td>6048</td>
</tr>
<tr>
<td>03 Message XFER record</td>
<td>30</td>
<td>72</td>
<td>2092</td>
</tr>
<tr>
<td>08 Message XFER record</td>
<td>24</td>
<td>64</td>
<td>1536</td>
</tr>
<tr>
<td>30 COMMIT record</td>
<td>202</td>
<td>44</td>
<td>8888</td>
</tr>
<tr>
<td>38 COMMIT record</td>
<td>37</td>
<td>44</td>
<td>1628</td>
</tr>
<tr>
<td>40 System checkpoint</td>
<td>487</td>
<td>45</td>
<td>403151</td>
</tr>
<tr>
<td>01 Begin checkpoint</td>
<td>1</td>
<td>628</td>
<td>628</td>
</tr>
<tr>
<td>03 CNT and/or LNT blocks</td>
<td>7</td>
<td>252</td>
<td>5868</td>
</tr>
<tr>
<td>04 SMB blocks</td>
<td>263</td>
<td>336</td>
<td>251856</td>
</tr>
<tr>
<td>05 CTB blocks</td>
<td>5</td>
<td>648</td>
<td>4488</td>
</tr>
<tr>
<td>06 DMB blocks</td>
<td>140</td>
<td>134</td>
<td>76862</td>
</tr>
<tr>
<td>07 PSB blocks</td>
<td>29</td>
<td>288</td>
<td>28512</td>
</tr>
<tr>
<td>08 CLB and/or LLB blocks</td>
<td>1</td>
<td>304</td>
<td>304</td>
</tr>
<tr>
<td>00 CCB blocks</td>
<td>32</td>
<td>484</td>
<td>29624</td>
</tr>
<tr>
<td>0F MSGQ TTR &amp; LCD</td>
<td>1</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>10 CRB blocks</td>
<td>1</td>
<td>744</td>
<td>744</td>
</tr>
<tr>
<td>21 VTCB blocks</td>
<td>5</td>
<td>144</td>
<td>1400</td>
</tr>
<tr>
<td>31 SIDX Information</td>
<td>1</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>98 End checkpoint info</td>
<td>1</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td>42 IMS started/Log EV reached</td>
<td>2</td>
<td>492</td>
<td>984</td>
</tr>
<tr>
<td>43 Log dataset control record</td>
<td>2</td>
<td>210</td>
<td>420</td>
</tr>
<tr>
<td>45 Checkpoint statistics record</td>
<td>15</td>
<td>156</td>
<td>3020</td>
</tr>
<tr>
<td>47 Active region record</td>
<td>1</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>48 Padding</td>
<td>73</td>
<td>46</td>
<td>3358</td>
</tr>
<tr>
<td>4C Data base started/stopped</td>
<td>9</td>
<td>28</td>
<td>252</td>
</tr>
<tr>
<td>50 Data base update/replace/delete</td>
<td>3148</td>
<td>111</td>
<td>505560</td>
</tr>
<tr>
<td>56 External subsystem TPCP</td>
<td>483</td>
<td>64</td>
<td>30976</td>
</tr>
<tr>
<td>00 IMS/V5 V1</td>
<td>6</td>
<td>72</td>
<td>576</td>
</tr>
<tr>
<td>07 Start of Unit-Of-Recovery (UOR)</td>
<td>236</td>
<td>64</td>
<td>15104</td>
</tr>
<tr>
<td>12 End of phase 2 syncpoint (DBCTL)</td>
<td>239</td>
<td>64</td>
<td>15296</td>
</tr>
<tr>
<td>66 DC enqueue for MSC, 3600 etc</td>
<td>663</td>
<td>48</td>
<td>31824</td>
</tr>
<tr>
<td>C5 User record</td>
<td>63</td>
<td>104</td>
<td>6352</td>
</tr>
<tr>
<td>C6 User record</td>
<td>114</td>
<td>104</td>
<td>11856</td>
</tr>
</tbody>
</table>

Figure 7. Example of R0 report
Using log and record procedures within IMS

DRL3IN1 (R1)

The output ddname for this record procedure is DRLIRPT1, when run under DRL2LOGP control.

The ddname should point to a data set with a fixed blocked (FB), logical record length of 133 bytes. The printable report output gives a detailed record of every transaction, BMP program, or message switch found on the input log processed.

Figure 8 shows an example of an R1 report. See “IMS_TRAN_TYPE key column” on page 81 for a complete description of the Transact type column (IMS_TRAN_TYPE) shown on this report.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Start</th>
<th>Pgm_Name</th>
<th>Region</th>
<th>In Q</th>
<th>Proc</th>
<th>OutQ</th>
<th>Tran</th>
<th>Net</th>
<th>CPU</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>F-----</td>
<td>MTDPRINT</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000002</td>
<td>F-----</td>
<td>MTDPRINT</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000003</td>
<td>F-----</td>
<td>MTDPRINT</td>
<td>0.0</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000004</td>
<td>FC-----</td>
<td>BMP99902</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000005</td>
<td>FD-----</td>
<td>PGM11000</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000006</td>
<td>FF-----</td>
<td>TRN91012</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000007</td>
<td>FE-----</td>
<td>TRN91031</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000008</td>
<td>FF-----</td>
<td>TRN91210</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000009</td>
<td>FF-----</td>
<td>TRN91210</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000010</td>
<td>FC-----</td>
<td>BMP99902</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000011</td>
<td>FC-----</td>
<td>BMP99902</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000012</td>
<td>EE-----</td>
<td>LTFF0024</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000013</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000014</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000015</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000016</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000017</td>
<td>EE-----</td>
<td>LTFF0024</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000018</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000019</td>
<td>FF-----</td>
<td>LTRM0005</td>
<td>IMSMPRT1</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DRL3IN2 (R2)

The output ddname for this record procedure is DRLIRPT2, when run under DRL2LOGP control.

The ddname should point to a variable blocked (VB) data set with a logical record length of at least 500 bytes. The output records are similar to those produced with R1, but they contain more data. The intended primary function of R2 is to reformat the composite records. Most of the data produced by R2 is in packed binary format and is not printable, but the log collector can use it as input to update the database.
Using log and record procedures within IMS
Chapter 3. Understanding data flow through IMS performance feature

This chapter describes the flow of data through the IMS Performance feature. The chapter includes:

- Tivoli Decision Support for OS/390 data flow overview
- IMS Performance feature data flow through the log and record procedures
  - When collecting data into DB2 tables using the Tivoli Decision Support for OS/390 log collector
  - When generating output composite records using the DRL2LOGP batch program
- IMS Performance feature data flow through the collect component

Overview of Tivoli Decision Support for OS/390 data flow

Figure 9 on page 24 shows an overview of the flow of data within Tivoli Decision Support for OS/390. It illustrates the steps involved in processing data.
The data flow follows these steps:

1. The operating system or other program writes data to a sequential log data set, which is the input to Tivoli Decision Support for OS/390.

2. You initiate the collect either through the dialog, or by using a Tivoli Decision Support for OS/390 language statement in a job, identifying a specific log type definition.

3. Optionally, the log definition might process the log data with a user-exit program, a log procedure. If the log definition calls a log procedure:

Figure 9. Overview of Tivoli Decision Support for OS/390 data flow
a. The log procedure receives each record in the log as input.
b. Output from a log procedure varies in format, and is usually a record mapped by a Tivoli Decision Support for OS/390 record definition.

4. Tivoli Decision Support for OS/390 looks for record definitions associated with the log definition in its system tables. It applies those record definitions to specific record types from the log or log procedure.

5. Optionally, a record definition might require processing by a user-exit program, a record procedure. If a record definition requires processing by a record procedure:
   a. The record procedure receives a specific record type only and is not called for other record types.
   b. Output from a record procedure varies in format, and is usually a record mapped by a Tivoli Decision Support for OS/390 record definition.

6. Tivoli Decision Support for OS/390 applies a specific update definition to each known record type and performs the data manipulations and database updates as specified.

7. Tivoli Decision Support for OS/390 often selects data from lookup tables to fulfill the data manipulations that update definitions require.

8. Tivoli Decision Support for OS/390 writes nonsummarized and first-level summarized data to data tables specified by the update definitions.

9. Tivoli Decision Support for OS/390 uses updated tables as input for updating other, similar tables that are for higher summary levels. If update definitions specify data summarization:
   a. Tivoli Decision Support for OS/390 selects data from the first table as required by the update definitions and performs required data summarization.
   b. Tivoli Decision Support for OS/390 updates other data tables as required by the update definitions.

(Tivoli Decision Support for OS/390 might select data from lookup tables during this process, but this step is not shown in Figure 9 on page 24.)

10. Once Tivoli Decision Support for OS/390 stores the data from a collect, you can display reports on the data. Tivoli Decision Support for OS/390 uses a QMF query to select the data for the report.

11. Optionally, Tivoli Decision Support for OS/390 might select data from lookup tables specified in the query.

12. Tivoli Decision Support for OS/390 creates report data, displaying, printing, or saving it as you requested.

---

**Log collector data flow**

Figure 10 on page 26 shows the flow of data from the SLDS to the DB2 tables when you use the log collector. For information about running the log collector, see “Running the log collector” on page 39.
The log collector goes through these steps to update the database tables:

1. The IMS log contains the original data as written by IMS. The IMS Performance feature uses it as input to the log collector. The ddname DRLLOG points to the data set you want to process.

2. The log collector calls the log procedure (DRL2innL, where \( m \) is the level of IMS you are using; for example, DRL2I61L for IMS/ESA Version 6 Release 1).

3. The log procedure calls a parameter-checking procedure to verify the processing options you have 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. At key commit times while the log collector is running, the log procedure writes to the checkpoint file DRLICHKO. You can use DRLICHKO 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. DRLICHKO becomes DRLICHKI the next time you use the log procedure.

   At the start of processing, the log procedure checks for the existence of data set 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.

5. 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, where the record procedure simplifies the records for easier collection and reporting.

7. The log collector uses the extract records from the record procedure, the composite records from the log procedure, and the original records to update the DB2 tables. See "IMS Performance feature data flow" on page 28 for information about data flows for table updates.
DRL2LOGP data flow

Figure 11 shows the flow of data from the SLDS to the output records when you use DRL2LOGP. For information about running DRL2LOGP, see “Using DRL2LOGP” on page 40.

DRL2LOGP goes through these steps to create output:

1. The IMS log contains the original data as written by IMS and the IMS Performance feature uses it as input to DRL2LOGP. The ddname is DRLLOG, which points to the data set you want to process.

2. DRL2LOGP calls a parameter-checking procedure to check the processing options you have selected. If you do not specify any parameters, DRL2LOGP uses the defaults. The ddname is DRLIPARM, which points to the parameter data set or contains in-stream parameters.

3. DRL2LOGP calls the log procedure (DRL2InnL, where nn is the level of IMS you are using; for example, DRL2I61L for IMS/ESA Version 6 Release 1).

4. The log procedure calls a parameter-checking procedure to verify the processing options you have 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. At key commit times while the log collector is running, the log procedure writes to the checkpoint file DRLICHKO. You can use DRLICHKO 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. DRLICHKO becomes 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. The log procedure does all of the matching and processing to create the composite records, and sends the composite records back to DRL2LOGP.

7. The DRL2LOGP sends the selected records to record procedures, which can simplify the records for easier collection and reporting. You can specify...
multiple record procedures, and the IMS Performance feature writes the output
to data set DRLIRPTx (where x corresponds to the name of the record
procedure; for example, DRL3Imn3 would write to DRLIRPT3.) You can specify
the output data set as disk, tape, or dummy output.

8. DRL2LOGP sends the composite records to data set DRLICOMP, which you
can specify as disk, tape, or dummy output.

**IMS Performance feature data flow**

Figure 12 and Figure 13 on page 29 show the flow of data through the IMS
Performance feature when you use the log collector. They show the data flow from
the original IMS log data to the IMS Performance feature reports. They give a more
inclusive picture of the IMS Performance feature data flow; compare them with
Figure 10 on page 26.

In the figures, the characters nn stand for the IMS release and can be 51, 61, or 71.
The characters nnn stand for the IMS release and can be 510, 610, or 710. All IMS
Performance feature table names have the prefix IMS_, but this prefix is not shown
in the figure due to space constraints.

**Figure 12. Data flow—statistics subcomponent**
Figure 13. Data flow—application, transaction, and system subcomponents
Chapter 4. Administering the IMS Performance feature

This chapter explains how to use the IMS Performance feature to process and collect IMS data. You can use the log collector program alone or you can use DRL2LOGP in batch mode followed by the log collector, if you want to collect the resulting data. For more information about Tivoli Decision Support for OS/390 administration, refer to the Administration Guide.

Specifying DRL2LOGP and log procedure parameters

You can specify several parameters to control the operation of DRL2LOGP and the log procedure. Table 3 and [Table 4 on page 31](#) list these parameters, with a brief description, the default, and a guideline as to the impact on collect performance.

You can specify these 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 Version 5 Release 1 is specified:

IMSVER=51

<table>
<thead>
<tr>
<th>Table 3. Parameter summary for DRL2LOGP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter name</td>
</tr>
<tr>
<td>CICSNAME</td>
</tr>
<tr>
<td>IMSVER</td>
</tr>
<tr>
<td>REPORTS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4. Parameter summary for the log procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter name</td>
</tr>
<tr>
<td>FASTPATH</td>
</tr>
</tbody>
</table>
### Table 4. Parameter summary for the log procedure (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Possible values</th>
<th>Default</th>
<th>Description</th>
<th>Performance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>FULLFUNC</td>
<td>ESS, FPSYNC, CONV, MSGSWITCH, BMP</td>
<td>If the FULLFUNC parameter is not specified, all types of full function transactions are processed. If FULLFUNC= is specified, all types of full function transactions denoted by the possible values for this parameter are ignored.</td>
<td>Controls which types of full function transactions are ignored.</td>
<td>Medium</td>
</tr>
<tr>
<td>IMSID</td>
<td>Valid subsystem IDs, up to 8 characters long</td>
<td>IMS</td>
<td>The IMS subsystem ID.</td>
<td>None</td>
</tr>
<tr>
<td>IMSIDCHECK</td>
<td>CONTINUE, FAIL</td>
<td>CONTINUE</td>
<td>Controls whether the log procedure continues after a mismatched IMSID.</td>
<td>None</td>
</tr>
<tr>
<td>MAXFREE</td>
<td>Whole numbers between 800 and max. system capacity</td>
<td>800</td>
<td>Sets the number of internal buffers for queueing incomplete transactions.</td>
<td>High</td>
</tr>
<tr>
<td>MAXOUTPUT</td>
<td>Whole numbers between 1 and 65536</td>
<td>5</td>
<td>Limits the number of MSGQ outputs that are matched for a MSGQ input UOR combination.</td>
<td>Low</td>
</tr>
<tr>
<td>MAXUOR</td>
<td>Whole numbers between 1 and 65536</td>
<td>5</td>
<td>Limits the number of syncpoints (UORs) that are matched for a batch program or stray UOR.</td>
<td>Low</td>
</tr>
<tr>
<td>MTOTRAFFIC</td>
<td>YES, NO</td>
<td>NO</td>
<td>Specifies whether master terminal operator (MTO) data should be processed.</td>
<td>Low</td>
</tr>
<tr>
<td>PASSLOGREC</td>
<td>YES, NO</td>
<td>YES</td>
<td>Specifies whether the original IMS log records are passed as output from the log procedure.</td>
<td>High</td>
</tr>
<tr>
<td>PSEUDOWFICHE</td>
<td>YES</td>
<td>None</td>
<td>Specifies for WFI programs whether the incomplete transactions have to be held in memory and then, if not completed, written to the DRLICHKO data set to be concatenated with the next IMS file that is collected.</td>
<td>High (decreases the overall collect time)</td>
</tr>
<tr>
<td>RECTYPE</td>
<td>Valid hex numbers from 00 to FF</td>
<td>FF</td>
<td>Record type of composite records created.</td>
<td>None</td>
</tr>
</tbody>
</table>
## Table 4. Parameter summary for the log procedure (continued)

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Possible values</th>
<th>Default</th>
<th>Description</th>
<th>Performance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOLUTION</td>
<td>NORMAL HIGH EXTENDED EXTREME</td>
<td>NORMAL</td>
<td>Specifies the resolution of the CICS recovery token.</td>
<td>Low</td>
</tr>
<tr>
<td>RESTARTCHECK</td>
<td>NO None</td>
<td>None</td>
<td>Specifies whether the collect restarts if the IMS restart record is found. The collect restarts form the first valid record after the IMS restart record.</td>
<td>Low</td>
</tr>
<tr>
<td>START</td>
<td>Valid values for yyyy-mm-dd-hh:mm:ss.t</td>
<td>Start of log</td>
<td>The IMS log date and time from which to start processing.</td>
<td>None</td>
</tr>
<tr>
<td>STOP</td>
<td>Valid values for yyyy-mm-dd-hh:mm:ss.t</td>
<td>End of log</td>
<td>The IMS log date and time at which to stop processing.</td>
<td>None</td>
</tr>
<tr>
<td>TABLEFLUSH</td>
<td>Whole numbers between 1 and 9999</td>
<td>None</td>
<td>Age limit (in seconds) of table entries. Entries older than this limit are flushed from the table when an IMS checkpoint is encountered or the storage limit is reached.</td>
<td>Medium</td>
</tr>
<tr>
<td>WRITE-PENDING</td>
<td>NO YES</td>
<td>NO</td>
<td>Controls whether pending table entries (those not completed at the end of the log procedure) are written as output. If the DRLICHKO ddname is present, this parameter is ignored.</td>
<td>Low</td>
</tr>
<tr>
<td>XRFNAME</td>
<td>Customers XRF system name</td>
<td>DFSRSENM</td>
<td>Specifies the customer’s XRF name.</td>
<td>None</td>
</tr>
<tr>
<td>XRF SYS</td>
<td>YES NO</td>
<td>NO</td>
<td>Specifies whether XRF system / non-XRF system.</td>
<td>None</td>
</tr>
</tbody>
</table>

The parameters for DRL2LOGP are:

**CICSNAME=CICS name for the system**

The user-specified name of the CICS that is connected to the IMS system on which Tivoli Decision Support for OS/390 is collecting. The default value is CICS.

**IMSVER=nn**

Specifies which release of IMS log data you are using as input, where \( nn \) is the version and release number. For example, if you are using IMS Version 6 Release 1 data, specify IMSVER=61. The default is 51, which specifies IMS Version 5 Release 1.

**REPORTS=Rat(xx-yy)**

Specifies which report programs (or record procedures) to run when the IMS Performance feature encounters a particular record type. \( Ra \) identifies the record procedure where \( a \) is the last alphanumeric character of the
record procedure name and \((xx-yy)\) is the record type. For example, to run DRL3imn1 when the IMS Performance feature encounters record type X'FF’, specify REPORT=R1(FF). The default is to produce no reports. If you specify R0 without a record type, then all record types pass to the report program. You can specify multiple reports with one REPORTS keyword, each separated by a comma.

The parameters for the log procedure are:

**FASTPATH**=aaa,bbb,ccc

Specifies the types of Fast Path transaction activity the log procedure ignores. This parameter allows you to improve performance by reducing the number of record types that are matched. You can specify multiple values:

- **ESS**
  - External subsystem records (type X'56') are not matched.
- **FFSYNC**
  - Full function syncpoint records (type X'37') are not matched.
- **MSGQ**
  - Message queue records (types X'03', X'35', X'31', and X'36') are not matched for Fast Path transactions that issue message queue outputs. If this value is specified, meaning that these message queue record types are not matched, subtypes X''xxX'EF' and X''xxX'ED' are not possible.

If the FASTPATH parameter is not specified, all types of Fast Path transactions and related activity are processed and matched. If FASTPATH= is specified, all types of Fast Path transactions denoted by the possible values are not matched.

**FULLFUNC**=aaa,bbb,ccc,ddd,eee

Specifies the types of full function transaction activity the log procedure ignores. This parameter allows you to improve performance by reducing the number of record types that are matched. You can specify multiple values:

- **ESS**
  - External subsystem records (type X'56') are not matched.
- **FPSYNC**
  - Fast Path syncpoint records (type X'593'\(n\)) are not matched.
- **CONV**
  - Conversation records (types X'11', X'12', and X'13') are not matched.
- **MSGSWITCH**
  - Message queue record type X'01' (destined for a CNT) and type X'03' (representing a system message) are excluded. If you do not specify this value, meaning that these message queue record types are included, larger numbers of subtypes X''xxX'FB' and X''xxX'FA' records will be present.
- **BMP**
  - Program schedule and termination records (types X'08' and X'07') for BMP regions are not matched.

If the FULLFUNC parameter is not specified, all types of full function transactions and related activity are processed and matched. If FULLFUNC= is specified, all types of full function transactions denoted by the possible values are not matched.

**IMSID**=xxxxxxxx

Specifies the IMS subsystem name the IMS Performance feature should use
until a X’42’ record is encountered on the log, where xxxxxxxx is the name of the IMS subsystem. For example, if your IMS subsystem name is IMS2, specify IMSID=IMS2.

**IMSIDCHECK=xxxxxxxx**
Specifies whether the log collector should stop or continue if a mismatch occurs between the IMSID specified with the parameter and the IMSID found in the X’42’ record. Possible values are FAIL and CONTINUE. If you specify CONTINUE and a mismatch occurs, the IMS Performance feature uses the IMSID specified with the parameter instead of the IMSID found in the X’42’ record. The default is CONTINUE.

**MAXFREE=nnnn**
Specifies the number of internal buffers that will be used to queue incomplete transactions. Do not set this parameter to a value which is less than 800. Values under 800 will cause queueing problems because of inadequate number of internal buffers. If queueing problems occur, increase the value in 50% increments until the problem is resolved. However a too large increase in the parameter value may result in excessive system memory being assigned to the internal buffers. Possible values are from 800 to max. system memory space. The default is 800.

**MAXOUTPUT=nnnnnn**
Specifies the number of MSGQ outputs to match to MSGQ input UOR combinations. This parameter controls this type of matching and therefore prevents the composite records from becoming too large and being arbitrarily truncated and flushed.

**MAXUOR=nnnnnn**
Specifies the number of unit of recoveries (UORs) to match to the PSB recovery token for batch tasks and stray UORs. This parameter controls UOR matching and therefore prevents the composite records from becoming too large and being arbitrarily truncated and flushed.

**MTOTRAFFIC=xxx**
Specifies whether to include MTO- and IMS-generated message switch activity when writing composite records. Possible values are YES and NO. The default is NO.

**PASSLOGREC=xxx**
Specifies whether the original IMS log records are passed as output from the log procedure. Specify NO to use the IMS Performance feature as described in this book, except that the R0 report will not be produced. Specify YES to be able to use your own Tivoli Decision Support for OS/390 update definitions based on IMS log records. Possible values are YES and NO. The default is YES.

**PSEUDOWFICHK=YES**
When you specify YES for WFI programs, the incomplete transactions will not be held in memory and then, if not completed, will not be written to the DRLICHKO data set to be concatenated with the next IMS file that is collected.

**RECTYPE=nn**
Specifies the record type for the composite records, where nn is the record type. For example, if you want the composite records to have type X’FA’, specify RECTYPE=FA. The default is FF.

**RESOLUTION=xxxxxxxx**
Specifies the resolution of the CICS recovery token. The IMS recovery
token can be produced either by the IMS system or by a CICS system feature. This parameter should be set according to the transaction rate of the system. Possible values are:

**NORMAL**
For systems with transaction rates of less than 0.9/sec, from the same CICS and with the same transaction code.

**HIGH**
For systems with transaction rates between 0.9 and 100/sec, from the same CICS and with the same transaction code.

**EXTENDED**
For systems with transaction rates between 100 and 200/sec, from the same CICS and with the same transaction code.

**EXTREME**
For systems with transaction rates of more than 200/sec, from the same CICS and with the same transaction code.

For example, if your system CIC1 generates 10 transactions/sec with the transaction code TRN1 and NORMAL (or default) has been specified for this parameter then the collect job will abend with code 0002 and display the message

DRL2IMSZ DRL2ICFR
DRL2IMSZ DRL2ICFR at DUPERKEYERT Duplicate key found

Do not specify EXTENDED or EXTREME unless it is absolutely essential. If you specify EXTENDED or EXTREME, then the collect must be run at least once every 18 hours. When data is collected from more than 18 hours the data may be corrupted or the collect abended. Some examples of setting the parameter are given below.

**Examples of RESOLUTION parameter settings**

<table>
<thead>
<tr>
<th>TOD</th>
<th>Stepping Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>Days</td>
</tr>
<tr>
<td>47</td>
<td>000</td>
</tr>
<tr>
<td>39</td>
<td>000</td>
</tr>
<tr>
<td>31</td>
<td>000</td>
</tr>
<tr>
<td>15</td>
<td>000</td>
</tr>
<tr>
<td>7</td>
<td>203</td>
</tr>
</tbody>
</table>

**RESTARTCHECK=NO**
If you specify No, when an IMS restart record is found, Tivoli Decision Support for OS/390 does not stop execution. A DB2 commit is performed for all transactions that are consistent and the IMS collects and restarts by using the valid records after the IMS restart record.

**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.

**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 the number of seconds that an entry can remain in log procedure.
internal tables. The log procedure removes entries older than this limit from the table when an IMS checkpoint is encountered or the storage limit is reached. If you do not specify a TABLEFLUSH value, table entries accumulate indefinitely and the tables are not flushed. For example, if you specify TABLEFLUSH=5 and the storage limit is reached, transaction entries that started more than 5 seconds before the date and time found in the last processed relevant record are removed from the table.

**WRITEPENDING=xxx**
Specifies whether the log procedure writes unmatched table entries to an output record when it is finished processing. If the DRLICHKO ddname is present, the log procedure ignores this parameter. Possible values are NO and YES. The default is NO.

**XRFRAME=xxxxxxxx**
Specifies the customers XRF system name. The default is DFSRSEMN.

**XRFSYS=xxx**
Specifies whether the recovery token is IMS or not. Possible values are NO and YES. The default is NO.
Specifying log collector parameters

When running the log collector, you specify log collector parameters and COLLECT statements using in-stream JCL or in a data set allocated to ddname DRLIN. The format for specifying these parameters is: the keyword SET in column 1, followed by the parameter name, an equal sign (=), the value in single quotes (’ ’), and ending with a semicolon (;). For example, the IMS_SYSTEM_ID of IMS1 is specified as:

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

Refer to the Language Guide and Reference for a description of the COLLECT and SET statements. Specify these parameters for IMS collect:

**MVS_SYSTEM_ID**

The ID for the MVS system where IMS was running. The IMS Performance feature uses this value as a key column in all tables, so specify it with care. If you do not specify this parameter, the column contains the value $UNKNOWN.

**IMS_SYSTEM_ID**

The ID for the IMS system. The IMS Performance feature uses this value as a key column in all tables, so specify it with care. If you do not specify this parameter, the column contains the value $UNKNOWN.

**IMS_APPLID**

The VTAM® APPLID for the IMS system. The IMS Performance feature does not use this value as a key column, but it is present in all tables and may be useful in cross analysis of IMS activity with network data. If you do not specify this parameter, the column contains the value $UNKNOWN.

**IMS_CTRL_REGION**

The JES and MVS recognized job name for the control region of the IMS system. The IMS Performance feature does not use this value as a key column, but it is present in all tables and may be useful in cross analysis of IMS activity with SMF data. If you do not specify this parameter, the column contains the value $UNKNOWN.
Running the log collector

You can run collect online using the administration dialog, or in batch. To run collect, refer to the Administration Guide.

A sample job for collecting IMS data is in member DRLJCOIM in the Tivoli Decision Support for OS/390 CNTL library. Figure 14 shows sample JCL that you can use to run the log collector in batch mode.

```plaintext
//USERIDA JOB (ACC000,001),'IMS1 COLLECT',
   // NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=OM
   //COLLECT EXEC PGM=DRLPLC,
   // PARM=('SYSTEM=DB21',
         // 'SYSPREFIX=DRLSYS',
         // 'BPREFIX=IMSPCTL',
         // 'SHOWSQL=NO',
         // 'SHOWINPUT=NO')
//STEPLIB DD DISP=SHR,DSN=Tivoli Decision Support for OS/390 load library
// DRLLOG DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS
// 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 *
IMSID=IMS1
IMSIDCHECK=FAIL
MAXOUTPUT=50
MAXUOR=50
/*
// DRLIN DD DISP=SHR,DSN=USERIDA.IMS.DEFS.V51(DRL$CVAR)
// DD *
COLLECT IMS_Vnnn_COLLECT -- IMS Vnnn
BUFFER SIZE 30000000 -- Appropriate collect buffer size
/*
```

Figure 14. Sample job for running the log collector

In Figure 14, the log collector parameters are specified using the PDS member DRLJ$CVAR. You can also specify them as in-stream parameters after a DRLIN dd statement. Figure 15 shows an example of a data set containing log collector parameters and a COLLECT statement for IMS.

```plaintext
***************************************************************************
/*
/* IMS log collector parameters used during data collection */
/*
***************************************************************************
SET MVS_SYSTEM_ID = 'MVSP' ;
SET IMS_SYSTEM_ID = 'IMS1' ;
SET IMS_APLID = 'IMSPCTL' ;
SET IMS_CTRL_REGION = 'PMISCTL' ;
COLLECT IMS_Vnnn_COLLECT ;
***************************************************************************
```

Figure 15. Sample data set with log collector parameters
Using DRL2LOGP

DRL2LOGP is a stand-alone batch program that you can use to run the IMS Performance feature log procedure in a non-Tivoli Decision Support for OS/390 environment. DRL2LOGP is another way to run the log procedure. It is the equivalent of running the log procedure under the log collector, except that DB2 is never invoked and DB2 tables are not updated. DRL2LOGP can create a large amount of output. You should use it only for detailed analysis.

DRL2LOGP input and output data sets

DRL2LOGP has these inputs and outputs, listed here by ddname:

**DRLLOG—input IMS logs**

The input IMS log data. The DSECT of the IMS log record type for the IMS version you are using describes the layout of each record. The input log is usually the SLDS or an appropriate extract, but you can use the OLDS after IMS has closed it. See Appendix D, “Sample archive exit”, on page 229 for an example of the archive utility exit to extract the record types required by the IMS Performance feature log procedure.

*Note:* Do not interfere with any normal IMS processing, including archiving of OLDS data.

**DRLICOMP—output composite records**

The output composite records. The DRLJXIDC macro in Appendix C, “DRLJXIDC DSECT macro”, on page 223 and the record definitions IMS_Vnnn_TRAN and IMS_Vnnn_STxxx describe the layout of each record. The IMS Performance feature writes the composite records to the data set associated with the ddname DRLICOMP. Either dummy this ddname, allocate it to a data set with record length of 32 756 bytes.

**DRLICHKI—input log procedure checkpoint file (optional)**

The input data set that contains the status of all pending IMS activity written when the log procedure completed processing on a previous run. This data set ensures that the IMS Performance feature can process IMS log data in discrete data set level parts without loss of data. The layout is internal. You can dummy this ddname, allocate it to a previously created output log procedure checkpoint file, or leave it out.

**DRLICHKO—output log procedure checkpoint file (optional)**

The output data set that records the status of all pending IMS activity 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. The layout is internal. Do not change it. This data set can be quite large for a large IMS system with many secondary transactions. You can dummy this ddname, allocate it to a data set with record length of 32 756 bytes, or leave it out.

**DRLOUT—output messages**

The IMS Performance feature writes messages to this ddname. You can allocate this ddname to SYSOUT, a physical data set, or dummy.

**DRLDUMP—output error information**

The IMS Performance feature writes error information to this ddname. You can allocate this ddname to SYSOUT, a physical data set, or dummy.

**DRLIRPTn—output from report Rn**

The IMS Performance feature writes report output to this ddname. You should allocate this ddname to a data set or SYSOUT, according to the
requirements of the particular report program. For example, DRL3Inn0 writes output to ddname DRLIRPT0, which should have a record length of 80 bytes. DRL3Inn1 writes to ddname DRLIRPT1, which should have a record length of 133 bytes.

**DRLIPARM—input log procedure parameters**
This ddname points to the parameter data set or contains the in-stream parameters. See Figure 1 on page 39 for an example of JCL specifying a data set containing the log procedure parameters. If you do not specify this ddname or specify it as DUMMY, the log procedure uses parameter defaults.

**Running DRL2LOGP**
DRL2LOGP reads log records from the input IMS log and invokes the IMS log procedure with each of them. It replicates the Tivoli Decision Support for OS/390 log collector functions and maintains the same interface with the log procedure. DRL2LOGP also calls report programs DLR3InnX (where \( n = \) the IMS release number and \( X = 0—9, A—Z \)). DRL2LOGP output goes to DRLICOMP and DRLIRPTX (where \( X \) corresponds to the name of the report program (or record procedure) that DRL2LOGP calls.

The report program DRL3Inn0 produces a report for IMS log data similar to that produced by the IFASMFDP program for SMF data. Program DRL3Inn1 produces a detailed transaction report similar to those available from existing IMS report programs DFSILTA0 and DBFULTA0 and the IMSPARS LOG report.

A sample job that can be used to run DRL2LOGP is in member DRLJLOGP in the Tivoli Decision Support for OS/390 CNTL library. Figure 16 shows an example job.

```plaintext
//USERIDA JOB (ACC000,001),'DRL2LOGP',
  NOTIFY=USERIDA,MSGCLASS=A,REGION=0M
  //DRL2LOGP EXEC PGM=DRL2LOGP
  //STEPLIB DD DISP=SHR,DSN=Tivoli Decision Support for OS/390 load library
  //DRLLOG DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS
  //DRLICOMP DD DUMMY -- or LRECL=32756 output composite data set
  //DRLICH1 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)
  //DRLIRPT0 DD SYSOUT=*,DCB=(LRECL=80)
  //DRLIRPT1 DD SYSOUT=*,DCB=(LRECL=133)
  //DRLIPARM DD *
  IMSVER=51 -- the default, or IMSVER=61, or IMSVER=71
  REPORTS=R0(00-FF)
  REPORTS=R1(FF)
  IMSID=IMS1
  IMSIDCHECK=FAIL
  MAXOUTPUT=50
  MAXUOR=50
  //
```

**Figure 16. Sample job for running DRL2LOGP**
Operational considerations

To make the log procedure and the collect process most effective, note these operational considerations when using them:

- Use the IMSIDCHECK=FAIL parameter to ensure that you are processing the correct IMS systems data.
- Limit the MAXUOR and MAXOUTPUT counts to approximately 50 each. If you specify a larger value to avoid composite record truncation, the resulting record is less useful.
- Use the TABLEFLUSH parameter sparingly. Acquiring more virtual storage to store pending nodes can guarantee correct and complete output.
- You can use the IMS archive exit utility to extract the necessary records from the IMS OLDS at archive time. This saves some processing, but because of the comprehensiveness of the IMS Performance feature, the savings of space and processing are much less than when using the archive exit utility with other products, such as SLR.
- Process logs from discrete IMS sessions or parts of an IMS session only.
- Do not attempt to process log data from different IMS sessions of the same IMS system that have been written to the same physical data set. The possibility of cold starts or emergency restarts in the middle of the data set may lead to duplicate key conditions.
- Do not attempt to process log data from different IMS systems in the same collect or DRL2LOGP process. Keys between two IMS systems may overlap and cause duplicate key conditions.
- Do not attempt to process across gaps in logs where data is missing. Matching keys may have been reused across the gap in the data, causing duplicate key conditions.
- Large IMS installations can use this method to collect their IMS log data into the IMS Performance feature tables:
  1. Set up your IMS collect job to use a generation data group or equivalent to manage your IMS Performance feature checkpoint files, reading relative generation (0) from DRLICHKI and writing relative generation (+1) to DRLICHKO.
  2. Allow OPC or an equivalent product to submit the collect job after successful completion of each IMS archive job, reading the latest SLDS created.

When using this method of collection, you may wish to allow page level locking to avoid any lockouts between users wishing to use the IMS tables and the IMS Performance feature COLLECT, which would attempt to update those same tables.

This method of collecting IMS data does not result in any loss of information, due to the use of the IMS Performance feature log procedure checkpoint facility. Although imposing a slightly greater operational demand, this method could enable you to more easily collect very large volumes of data.
Recovering from abends during collect

The IMS Performance feature log procedure saves information from the input records and combines this information to produce composite records. However, Tivoli Decision Support for OS/390's 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 Performance 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 Performance feature log procedure checkpoint facility, be careful 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 done as follows.

- If the IMS Performance feature DB2 tables were not updated before the failure, rerun the collect job after correcting the cause of the error.
- If the IMS Performance feature DB2 tables were updated before the failure, restore the tables to the status before the collect job that failed. This can be done from DB2 backup copies of the IMS Performance Feature tables. Run the DB2 RECOVER utility, using standard DB2 procedures for point-in-time recovery. Refer to the DB2 Administration Guide: Volume 2 and Volume 3 for information about DB2 recovery.

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

Additional capabilities

The following information is not stored in the DB2 tables provided with the IMS Performance feature, but is available for processing into user-defined tables. See the source for the IMS_Vnnn_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
Administering the IMS Performance feature
Chapter 5. IMS performance feature log and record definitions

This chapter describes:

- IMS Performance feature log definitions
- IMS Performance feature record definitions corresponding to IMS record types
- Composite and DRL3I mn2 extract (R2 extract) record definitions as created by the IMS Performance feature log and record procedure

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

Log definitions

Log definitions reside in the Tivoli Decision Support for OS/390 system tables. They define each log to the IMS Performance feature. A log must be defined to Tivoli Decision Support for OS/390 before any data can be collected. You specify the log definition that you want to use in the COLLECT statement. Refer to the Administration Guide for more information about using the COLLECT statement.

Depending on which components you install, one or more of these log definitions will be installed:

**IMS_Vnnn_COLLECT**

This log definition is part of the collect components. It works together with the IMS Performance feature log procedures and record procedures. You use it for standard Tivoli Decision Support for OS/390 data collection by issuing the COLLECT IMS_Vnnn_COLLECT statement.

**IMS_Vnnn_SLDS**

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

Record definitions

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

Comparison of performance programs

Table 5 compares the IMS records used by the IMS Performance feature with those used by similar performance products and programs. The products and programs are abbreviated in the table:

<table>
<thead>
<tr>
<th>Product</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DFSILTA0</td>
<td>IMS Log Transaction Analysis utility</td>
</tr>
<tr>
<td>DFSULTA0</td>
<td>IMS Fast Path Log Analysis utility</td>
</tr>
<tr>
<td>IMSPARS</td>
<td>IMS Performance Analysis and Reporting System</td>
</tr>
<tr>
<td>SLR 3.3</td>
<td>Service Level Reporter Version 3 Release 3</td>
</tr>
<tr>
<td>Tivoli Decision Support for OS/390/IMS</td>
<td>IMS Performance feature</td>
</tr>
</tbody>
</table>

Table 5. Comparison of the IMS Performance feature with other products for IMS record types

<table>
<thead>
<tr>
<th>Record</th>
<th>Description</th>
<th>DFSILTA0</th>
<th>DFSULTA0</th>
<th>IMSPARS</th>
<th>SLR 3.3</th>
<th>Tivoli Decision Support for OS/390/IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>Message received from a CNT</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'03'</td>
<td>Message received from DL/I</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'07'</td>
<td>Program termination</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'08'</td>
<td>Program initiation</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X'11'</td>
<td>Start of conversation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'12'</td>
<td>End of conversation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'13'</td>
<td>SPA record</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'31'</td>
<td>Message queue GU</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'32'</td>
<td>Message queue reject</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X'33'</td>
<td>Message queue free</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'34'</td>
<td>Message cancel</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'35'</td>
<td>Message queue enqueue</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'36'</td>
<td>Message queue dequeue</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'37'</td>
<td>Syncpoint record</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'38'</td>
<td>Message after abend</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'4001'</td>
<td>IMS checkpoint begin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'4004'</td>
<td>Checkpoint SMB</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'4098'</td>
<td>IMS checkpoint end</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'42'</td>
<td>Log buffer control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'45'</td>
<td>Statistics records</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'47'</td>
<td>Active region</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'56'</td>
<td>External subsystem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5901'</td>
<td>Fast Path input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5903'</td>
<td>Fast Path output</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Table 5. Comparison of the IMS Performance feature with other products for IMS record types (continued)

<table>
<thead>
<tr>
<th>Record</th>
<th>Description</th>
<th>DFSILTA0</th>
<th>DFSULTA0</th>
<th>IMSPARS</th>
<th>SLR 3.3</th>
<th>Tivoli Decision Support for OS/390/IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'5936'</td>
<td>Fast Path dequeue</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5937'</td>
<td>Fast Path syncpoint</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5938'</td>
<td>Fast Path abend</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

**Descriptions of record definitions**

Record definition names for IMS follow this general format:

**IMS**
Common prefix that indicates an IMS record definition

**Vnnn**
Indicates the release number (for example V510 signifies IMS Version 5 Release 1)

**record type**
Corresponding IMS record type

Table 6 on page 47 lists IMS record types with the corresponding IMS Performance 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 6. IMS record types and IMS Performance feature record definitions

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>IMS_Vnnn_01</td>
<td>Message queue record (message received from a CNT).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This record represents the message and its text and control information as it appears on the IMS message queues.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The DRRN indicates the message queue type:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X’00.....’ indicates QBLKS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X’04.....’ indicates short message queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X’08.....’ indicates long message queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The record contains the indicator of the origin of the message, its destination, whether it uses MSC, and so on.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>X'02'</td>
<td>IMS_Vnnn_02</td>
<td>Command log record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>
Log and record definitions

Table 6. IMS record types and IMS Performance feature record definitions  (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
</table>
| X'03'       | IMS_Vnnn_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_Vnnn_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 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'07'</td>
<td>IMS_Vnnn_07</td>
<td>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 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.</td>
</tr>
<tr>
<td>X'08'</td>
<td>IMS_Vnnn_08</td>
<td>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.</td>
</tr>
<tr>
<td>X'09'</td>
<td></td>
<td>Sequential buffering statistics.</td>
</tr>
<tr>
<td>X'0A07'</td>
<td>IMS_Vnnn_0A07</td>
<td>CPI-C program termination.</td>
</tr>
<tr>
<td>X'0A08'</td>
<td>IMS_Vnnn_0A08</td>
<td>CPI-C program initialization.</td>
</tr>
<tr>
<td>X'0F'</td>
<td></td>
<td>LE DECB record.</td>
</tr>
<tr>
<td>X'10'</td>
<td>IMS_Vnnn_10</td>
<td>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.</td>
</tr>
<tr>
<td>Record type</td>
<td>Record definition</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>X'11'</td>
<td>IMS_Vnnn_11</td>
<td>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.</td>
</tr>
<tr>
<td>X'12'</td>
<td>IMS_Vnnn_12</td>
<td>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.</td>
</tr>
<tr>
<td>X'13'</td>
<td>IMS_Vnnn_13</td>
<td>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 zeroes. 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.</td>
</tr>
<tr>
<td>X'14'</td>
<td></td>
<td>Switched-line disconnect.</td>
</tr>
<tr>
<td>X'15'</td>
<td></td>
<td>Switched-line connect.</td>
</tr>
<tr>
<td>X'16'</td>
<td>IMS_Vnnn_16</td>
<td>Sign-on/off record. This record logs the security maintenance utility (SMU) or Resource Access Control Facility (RACF®) user sign-on and sign-off.</td>
</tr>
<tr>
<td>X'18'</td>
<td>IMS_Vnnn_18</td>
<td>Extended checkpoint record. This record logs the details and checkpoint data for an extended checkpoint.</td>
</tr>
<tr>
<td>X'20'</td>
<td>IMS_Vnnn_20</td>
<td>Database open record. This record indicates that a DL/I database was opened and describes various key characteristics of the database and its files.</td>
</tr>
<tr>
<td>X'21'</td>
<td>IMS_Vnnn_21</td>
<td>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.</td>
</tr>
<tr>
<td>X'24'</td>
<td>IMS_Vnnn_24</td>
<td>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 record (CCHHR) details of the error.</td>
</tr>
<tr>
<td>X'25'</td>
<td></td>
<td>EEQE record.</td>
</tr>
<tr>
<td>X'26'</td>
<td></td>
<td>I/O toleration buffer.</td>
</tr>
<tr>
<td>X'27'</td>
<td></td>
<td>Database extension.</td>
</tr>
<tr>
<td>X'28'</td>
<td></td>
<td>Phase 1 DC record.</td>
</tr>
</tbody>
</table>
Table 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’30’</td>
<td>IMS_Vnnn_30</td>
<td>Message queue prefix changed record. This record logs changes made to the message queue record prefix. It links directly to a previously logged message.</td>
</tr>
<tr>
<td>X’31’</td>
<td>IMS_Vnnn_31</td>
<td>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.</td>
</tr>
<tr>
<td>X’32’</td>
<td>IMS_Vnnn_32</td>
<td>Message queue reject record. This record is produced when the MSGQ rejects a message because an error occurred, presumably causing an application program abend.</td>
</tr>
<tr>
<td>X’33’</td>
<td>IMS_Vnnn_33</td>
<td>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.</td>
</tr>
<tr>
<td>X’34’</td>
<td>IMS_Vnnn_34</td>
<td>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.</td>
</tr>
<tr>
<td>X’35’</td>
<td>IMS_Vnnn_35</td>
<td>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.</td>
</tr>
</tbody>
</table>

Log and record definitions

Chapter 5. IMS performance feature log and record definitions 51
### Log and record definitions

Table 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'36'</td>
<td>IMS_Vnnn_36</td>
<td>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.</td>
</tr>
<tr>
<td>X'37'</td>
<td>IMS_Vnnn_37</td>
<td>Message queue syncpoint transfer record. This record indicates that the message transferred to the permanent destination and reflects that a successful commit occurred.</td>
</tr>
<tr>
<td>X'38'</td>
<td>IMS_Vnnn_38</td>
<td>Message queue syncpoint fail record. This record indicates that a syncpoint failure occurred and the message transfer will not occur.</td>
</tr>
<tr>
<td>X'39'</td>
<td></td>
<td>Cleanup outqueue release.</td>
</tr>
<tr>
<td>X'3A'</td>
<td></td>
<td>DFSQFIX0 free.</td>
</tr>
<tr>
<td>X'3B'</td>
<td></td>
<td>DFSQFIX0 invalid message.</td>
</tr>
<tr>
<td>X'3C'</td>
<td></td>
<td>DFSQFIX0 validity check.</td>
</tr>
<tr>
<td>X'3D'</td>
<td></td>
<td>DFSQFIX0 QBLK altered.</td>
</tr>
<tr>
<td>X'3E'</td>
<td></td>
<td>Message chain update.</td>
</tr>
<tr>
<td>X'4001'</td>
<td>IMS_Vnnn_4001</td>
<td>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.</td>
</tr>
<tr>
<td>X'4002'</td>
<td>IMS_Vnnn_4002</td>
<td>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.</td>
</tr>
<tr>
<td>X'4003'</td>
<td>IMS_Vnnn_4003</td>
<td>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.</td>
</tr>
<tr>
<td>X'4004'</td>
<td>IMS_Vnnn_4004</td>
<td>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.</td>
</tr>
<tr>
<td>X'4005'</td>
<td>IMS_Vnnn_4005</td>
<td>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.</td>
</tr>
<tr>
<td>X'4006'</td>
<td>IMS_Vnnn_4006</td>
<td>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.</td>
</tr>
</tbody>
</table>
Table 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X’4007’</td>
<td>IMS_Vnnn_4007</td>
<td>Checkpoint PSB record. This record contains a checkpoint of all the PSBs defined to the IMS system at the time of the checkpoint.</td>
</tr>
<tr>
<td>X’4008’</td>
<td>IMS_Vnnn_4008</td>
<td>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.</td>
</tr>
<tr>
<td>X’4009’</td>
<td></td>
<td>Checkpoint CPT.</td>
</tr>
<tr>
<td>X’400A’</td>
<td></td>
<td>Checkpoint CPM.</td>
</tr>
<tr>
<td>X’400B’</td>
<td></td>
<td>Checkpoint CTM.</td>
</tr>
<tr>
<td>X’400C’</td>
<td></td>
<td>Checkpoint CVB.</td>
</tr>
<tr>
<td>X’400D’</td>
<td>IMS_Vnnn_400D</td>
<td>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.</td>
</tr>
<tr>
<td>X’400E’</td>
<td>IMS_Vnnn_400E</td>
<td>Checkpoint SPA record. This record contains a checkpoint of the currently allocated SPAs for active conversations.</td>
</tr>
<tr>
<td>X’400F’</td>
<td></td>
<td>Checkpoint LCB.</td>
</tr>
<tr>
<td>X’4010’</td>
<td></td>
<td>Checkpoint CRB.</td>
</tr>
<tr>
<td>X’4011’</td>
<td></td>
<td>Checkpoint TCM.</td>
</tr>
<tr>
<td>X’4014’</td>
<td>IMS_Vnnn_4014</td>
<td>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.</td>
</tr>
<tr>
<td>X’4015’</td>
<td>IMS_Vnnn_4015</td>
<td>Checkpoint EQE record. This record contains a checkpoint of all the generated error queue elements (EQEs) at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X’4020’</td>
<td>IMS_Vnnn_4020</td>
<td>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.</td>
</tr>
<tr>
<td>X’4021’</td>
<td>IMS_Vnnn_4021</td>
<td>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.</td>
</tr>
<tr>
<td>X’4025’</td>
<td></td>
<td>Checkpoint EEQE.</td>
</tr>
<tr>
<td>X’4026’</td>
<td></td>
<td>Checkpoint IEEQE/virtual I/O buffer.</td>
</tr>
<tr>
<td>X’4027’</td>
<td></td>
<td>In-doubt extended error queue elements (IEEQE).</td>
</tr>
<tr>
<td>X’4028’</td>
<td></td>
<td>Error queue elements (EQEL) for RIS.</td>
</tr>
<tr>
<td>X’4030’</td>
<td></td>
<td>Checkpoint SID.</td>
</tr>
<tr>
<td>X’4031’</td>
<td></td>
<td>Checkpoint RRE.</td>
</tr>
</tbody>
</table>
## Table 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'4070'</td>
<td>IMS_Vnnn_4070</td>
<td>Checkpoint MSDB begin record. This record contains system-wide information about IMS MSDBs, such as at the beginning of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4071'</td>
<td>IMS_Vnnn_4071</td>
<td>Checkpoint MSDB ECNT record. This record contains main storage database (MSDB) ECNT data for the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4072'</td>
<td>IMS_Vnnn_4072</td>
<td>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.</td>
</tr>
<tr>
<td>X'4073'</td>
<td>IMS_Vnnn_4073</td>
<td>Checkpoint MSDB pagefixed record. This record contains the checkpointed pagefixed MSDBs at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4074',</td>
<td>IMS_Vnnn_4074</td>
<td>Checkpoint MSDB pageable record. This record contains the checkpointed pageable MSDBs at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4079'</td>
<td>IMS_Vnnn_4079</td>
<td>Checkpoint MSDB end record. This record indicates that the IMS MSDB checkpoint process is now complete for this IMS checkpoint.</td>
</tr>
<tr>
<td>X'4080'</td>
<td>IMS_Vnnn_4080</td>
<td>Checkpoint Fast Path begin record. This record contains system-wide information about IMS Fast Path, such as at the beginning of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4081'</td>
<td>IMS_Vnnn_4081</td>
<td>Checkpoint Fast Path ECNT record. This record contains the checkpointed Fast Path ECNT data at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4082'</td>
<td>IMS_Vnnn_4082</td>
<td>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.</td>
</tr>
<tr>
<td>X'4083'</td>
<td>IMS_Vnnn_4083</td>
<td>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.</td>
</tr>
<tr>
<td>X'4084'</td>
<td>IMS_Vnnn_4084</td>
<td>Checkpoint FP DMCB/DMAC.</td>
</tr>
<tr>
<td>X'4085'</td>
<td>IMS_Vnnn_4085</td>
<td>Checkpoint Fast Path MTO buffer record.</td>
</tr>
<tr>
<td>X'4086'</td>
<td>IMS_Vnnn_4086</td>
<td>Checkpoint Fast Path DMHR/DEDB record.</td>
</tr>
<tr>
<td>X'4087'</td>
<td>IMS_Vnnn_4087</td>
<td>Checkpoint Fast Path ADSC record.</td>
</tr>
<tr>
<td>X'4088'</td>
<td>IMS_Vnnn_4088</td>
<td>Checkpoint Fast Path IEEQE record.</td>
</tr>
<tr>
<td>X'4089'</td>
<td>IMS_Vnnn_4089</td>
<td>Checkpoint Fast Path end record. This record indicates that the IMS Fast Path checkpoint process is now complete for this IMS checkpoint.</td>
</tr>
<tr>
<td>Record type</td>
<td>Record definition</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>X'4098'</td>
<td>IMS_Vnnn_4098</td>
<td>Checkpoint end blocks record. This record indicates that the IMS simple checkpoint is now complete.</td>
</tr>
<tr>
<td>X'4099'</td>
<td>IMS_Vnnn_4099</td>
<td>Checkpoint end queues record. This record indicates that the IMS dumpq checkpoint is now complete.</td>
</tr>
<tr>
<td>X'41'</td>
<td>IMS_Vnnn_41</td>
<td>Checkpoint batch record. This record indicates that a batch program has issued a checkpoint.</td>
</tr>
<tr>
<td>X'42'</td>
<td>IMS_Vnnn_42</td>
<td>Log buffer control record. This record indicates the status of IMS at log buffer end of volume and switch times.</td>
</tr>
<tr>
<td>X'43'</td>
<td>IMS_Vnnn_43</td>
<td>Log data set control record. This record indicates the status of the IMS OLDS data sets.</td>
</tr>
<tr>
<td>X'4502'</td>
<td>IMS_Vnnn_4502</td>
<td>Queue pool statistics record. This record contains statistics about the use of the message queue pool at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4503'</td>
<td>IMS_Vnnn_4503</td>
<td>Format buffer pool statistics record. This record contains statistics about the usage of the format buffer pool at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4504'</td>
<td>IMS_Vnnn_4504</td>
<td>Database buffer pool statistics record. This record contains statistics about the usage of the database buffer pool at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4505'</td>
<td>IMS_Vnnn_4505</td>
<td>Main pools statistics record. This record contains statistics about the usage of the principal pools at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4506'</td>
<td>IMS_Vnnn_4506</td>
<td>Scheduling statistics record. This record contains statistics about scheduling conflicts in IMS at the time of the checkpoint.</td>
</tr>
<tr>
<td>X'4507'</td>
<td>IMS_Vnnn_4507</td>
<td>Logger statistics record. This record contains statistics about the logical logger function of IMS at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4508'</td>
<td>IMS_Vnnn_4508</td>
<td>VSAM subpool statistics record. This record contains statistics about the VSAM subpools at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'4509'</td>
<td>IMS_Vnnn_4509</td>
<td>Program isolation statistics record. This record contains statistics about IMS program isolation and enqueue/dequeue at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'450A'</td>
<td>IMS_Vnnn_450A</td>
<td>Latch statistics record. This record indicates the status of IMS latches at checkpoint time.</td>
</tr>
</tbody>
</table>
### Log and record definitions

Table 6. IMS record types and IMS Performance feature record definitions (continued)

<table>
<thead>
<tr>
<th>Record type</th>
<th>Record definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'450B'</td>
<td>IMS_Vnnn_450B</td>
<td>Dispatch storage statistics record. This record indicates the selective dispatching storage pool status at the time of the checkpoint.</td>
</tr>
<tr>
<td>X'450C'</td>
<td>IMS_Vnnn_450C</td>
<td>DFSCBT00 storage statistics record. This record indicates the status of miscellaneous IMS storage pools at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'450D'</td>
<td>IMS_Vnnn_450D</td>
<td>RECANY (receive any) buffer statistics. This record contains statistics about the VTAM receive any (RECANY) buffer usage at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'450E'</td>
<td>IMS_Vnnn_450E</td>
<td>Storage manager statistics record.</td>
</tr>
<tr>
<td>X'450F'</td>
<td>IMS_Vnnn_450F</td>
<td>Dispatch statistics record.</td>
</tr>
<tr>
<td>X'45FF'</td>
<td>IMS_Vnnn_45FF</td>
<td>Statistics.</td>
</tr>
<tr>
<td>X'47'</td>
<td>IMS_Vnnn_47</td>
<td>Active region statistics record. This record contains information about all active regions, including BMP programs, at the time of the IMS checkpoint.</td>
</tr>
<tr>
<td>X'48'</td>
<td>IMS_Vnnn_48</td>
<td>OLDS padding record. This record contains padding and control information for the IMS OLDS.</td>
</tr>
<tr>
<td>X'4C'</td>
<td>IMS_Vnnn_4C</td>
<td>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.</td>
</tr>
<tr>
<td>X'5050'</td>
<td>IMS_Vnnn_5050</td>
<td>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.</td>
</tr>
<tr>
<td>X'5051'</td>
<td>IMS_Vnnn_5051</td>
<td>Full function database update unsuccessful record. This record indicates that the update action indicated by the previous X'50' record was unsuccessful.</td>
</tr>
<tr>
<td>X'5052'</td>
<td>IMS_Vnnn_5052</td>
<td>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.</td>
</tr>
<tr>
<td>X'53'</td>
<td></td>
<td>CI/CA and space manager.</td>
</tr>
<tr>
<td>X'55'</td>
<td>IMS_Vnnn_55FE0001</td>
<td>External subsystem DB2 snap in-doubt record. This record indicates that a DB2 external subsystem had to resolve in-doubt structures for a database.</td>
</tr>
<tr>
<td>X'56'</td>
<td>IMS_Vnnn_56</td>
<td>External subsystem record. This record indicates the status of external subsystem connection and commit processing.</td>
</tr>
<tr>
<td>Record type</td>
<td>Record definition</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>X'5901'</td>
<td>IMS_Vnnn_5901</td>
<td>Fast Path input message. This record indicates the receipt of an input message to the expedited message handler buffer for Fast Path processing.</td>
</tr>
<tr>
<td>X'5903'</td>
<td>IMS_Vnnn_5903</td>
<td>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.</td>
</tr>
<tr>
<td>X'5920'</td>
<td>IMS_Vnnn_5920</td>
<td>Fast Path MSDB change record. This record indicates the changing of an MSDB.</td>
</tr>
<tr>
<td>X'5921'</td>
<td>IMS_Vnnn_5921</td>
<td>Fast Path DEDB area data set open record. This record indicates the opening of a Fast Path DEDB area data set.</td>
</tr>
<tr>
<td>X'5922'</td>
<td>IMS_Vnnn_5922</td>
<td>Fast Path DEDB area data set close record. This record indicates the closing of a Fast Path DEDB area data set.</td>
</tr>
<tr>
<td>X'5923'</td>
<td>IMS_Vnnn_5923</td>
<td>Fast Path DEDB area data set status record. This record indicates the status of a Fast Path DEDB area data set.</td>
</tr>
<tr>
<td>X'5924'</td>
<td>IMS_Vnnn_5924</td>
<td>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.</td>
</tr>
<tr>
<td>X'5936'</td>
<td>IMS_Vnnn_5936</td>
<td>Fast Path dequeue message record. This record indicates that an expedited message handler message has been sent and successfully received by its destination node.</td>
</tr>
<tr>
<td>X'5937'</td>
<td>IMS_Vnnn_5937</td>
<td>EMH Fast Path syncpoint record. This record indicates that a successful Fast Path syncpoint occurred, indicating that any messages can be transmitted.</td>
</tr>
<tr>
<td>X'5938'</td>
<td>IMS_Vnnn_5938</td>
<td>EMH Fast Path syncpoint failure record. This record indicates that a Fast Path syncpoint failed and that message transmission may not occur.</td>
</tr>
<tr>
<td>X'5942'</td>
<td></td>
<td>Fast Path DMHR dequeue.</td>
</tr>
<tr>
<td>X'5947'</td>
<td></td>
<td>Fast Path MSSP image copy.</td>
</tr>
<tr>
<td>X'5950'</td>
<td>IMS_Vnnn_5950</td>
<td>Fast Path DEDB database update record. This record indicates that a Fast Path database online update occurred.</td>
</tr>
<tr>
<td>X'5951'</td>
<td>IMS_Vnnn_5951</td>
<td>Fast Path DEDB database update record. This record indicates an update made in a non-recoverable AREA/DEDB.</td>
</tr>
<tr>
<td>X'5953'</td>
<td>IMS_Vnnn_5953</td>
<td>Fast Path DEDB database update (utilities) record. This record indicates that Fast Path database utilities update occurred.</td>
</tr>
<tr>
<td>X'5954'</td>
<td>IMS_Vnnn_5954</td>
<td>Fast Path DEDB database open record. This record indicates the opening of a Fast Path DEDB database.</td>
</tr>
<tr>
<td>Record type</td>
<td>Record definition</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>X'5955'</td>
<td>IMS_Vnnn_5955</td>
<td>Fast Path sequential dependent syncpoint record. This record indicates that a new sequential dependent buffer was obtained during syncpoint processing.</td>
</tr>
<tr>
<td>X'5957'</td>
<td>IMS_Vnnn_5957</td>
<td>Fast Path database DMAC record.</td>
</tr>
<tr>
<td>X'5970'</td>
<td>IMS_Vnnn_5970</td>
<td>Fast Path hot standby MSDB relocation record. This record indicates that an MSDB has been relocated to the XRF hot standby system during takeover.</td>
</tr>
<tr>
<td>X'5B'</td>
<td>Buffered Fast Path.</td>
<td></td>
</tr>
<tr>
<td>X'5E'</td>
<td>Image capture of SB handler.</td>
<td></td>
</tr>
<tr>
<td>X'5F'</td>
<td>DL/I call trace.</td>
<td></td>
</tr>
<tr>
<td>X'62'</td>
<td>OSAM error.</td>
<td></td>
</tr>
<tr>
<td>X'63'</td>
<td>Allocate/deallocate.</td>
<td></td>
</tr>
<tr>
<td>X'64'</td>
<td>Message discarded by MSC.</td>
<td></td>
</tr>
<tr>
<td>X'65'</td>
<td>IRSS and SNA restart.</td>
<td></td>
</tr>
<tr>
<td>X'66'</td>
<td>3600 standard record.</td>
<td></td>
</tr>
<tr>
<td>X'67'</td>
<td>IMS_Vnnn_67</td>
<td>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.</td>
</tr>
<tr>
<td>X'67FA'</td>
<td>IMS_Vnnn_67FA</td>
<td>Trace table log record. This record contains the IMS trace table data.</td>
</tr>
<tr>
<td>X'69'</td>
<td>3275 switched unauthorized ID.</td>
<td></td>
</tr>
<tr>
<td>X'6E'</td>
<td>MSC link connect/disconnect.</td>
<td></td>
</tr>
<tr>
<td>X'6D'</td>
<td>XRF hot standby surveillance.</td>
<td></td>
</tr>
<tr>
<td>X'6E'</td>
<td>XRF session miscellaneous.</td>
<td></td>
</tr>
<tr>
<td>X'70'</td>
<td>Online change.</td>
<td></td>
</tr>
<tr>
<td>X'71'</td>
<td>TCF record.</td>
<td></td>
</tr>
<tr>
<td>X'7201'</td>
<td>IMS_Vnnn_7201</td>
<td>ETO user structure dynamically created.</td>
</tr>
<tr>
<td>X'7202'</td>
<td>IMS_Vnnn_7202</td>
<td>ETO user structure dynamically created.</td>
</tr>
<tr>
<td>X'7203'</td>
<td>IMS_Vnnn_7203</td>
<td>ETO user structure modified.</td>
</tr>
<tr>
<td>X'7204'</td>
<td>IMS_Vnnn_7204</td>
<td>CNT added to an ETO user structure.</td>
</tr>
</tbody>
</table>
Composite record definitions

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

**IMS_Vnnn_TRAN**

Composite transaction, message, and program task records.

Record types are: X'FF', X'FE', X'FD', X'FC', X'FB', X'FA', X'EF', X'EE', X'ED', X'EC', where xx is the record type, as set by the log procedure parameter RECTYPE=xx. The default is X'FF' if not explicitly set.

This record is created using the log records component.

**IMS_Vnnn_R2**

Composite transaction, after being simplified by the record procedure.

Record types are: X'FF', X'FE', X'FD', X'FC', X'FB', X'FA', X'EF', X'EE', X'ED', X'EC'.

This record is created using the collect component.

**IMS_Vnnn_STxxxx**

Statistics record, where xxxx is the record type: 4001, 4502, 4503, 4504, 4505, 4506, 4507, 4508, 4509, 450A, 450B, 450C, 450D, 450E, 47.

This record is created using the collect component.

### Composite record sections in IMS_Vnnn_TRAN

For maximum user flexibility, composite records are composed of all fields from the source records. The only exceptions to this are the records that normally contain user text strings, namely these record types:

- X'01'
- X'03'
- X'13'
- X'5901'
- X'5903'

In these cases, all fields *except* the text strings are transferred to the appropriate output composite record.

Each section of the composite record has a different name and prefix. Table 7 on page 60 explains the sections of the composite record:

<table>
<thead>
<tr>
<th>Short description</th>
<th>Name</th>
<th>Prefix</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>A descriptive section name</td>
<td>The actual name of the section</td>
<td>The one-character prefix for all data subsections and fields defined in the record section</td>
<td>A detailed explanation of the section and the data grouped in it, including the IMS log record types that compose it and any other notes that apply to the record section</td>
</tr>
</tbody>
</table>
### Table 7. Composite record sections in IMS_Vnnn_TRAN

<table>
<thead>
<tr>
<th>Short description</th>
<th>Name</th>
<th>Prefix</th>
<th>Section contents and explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program-to-program switch</td>
<td>ROOT</td>
<td>R</td>
<td>X'01'/X'03' Message queue insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'35' Message queue enqueue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'31' Message queue GU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the original message that started the program-to-program switch sequence of transactions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The same ROOT input message is added to all subsequent transactions in a switch sequence, regardless of whether transaction A started B, C and D, or transaction A started B, which started C, which then started D.</td>
</tr>
<tr>
<td>Message queue input</td>
<td>D1</td>
<td>I</td>
<td>X'01'/X'03' Message queue insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'35' Message queue enqueue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'31' Message queue GU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'36' Message queue dequeue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the input message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> If the composite record type and subtype is XFA’, only this key section is present in the record. This situation implies that a message switch has occurred; for example, one user may be sending a message to another user, or an MSC/ISC/FES message switch has occurred in the originating system.</td>
</tr>
<tr>
<td>Conversation</td>
<td>SPA</td>
<td>S</td>
<td>X'11' Conversation started</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'12' Conversation ended</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'13' Conversation SPA insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence as appropriate. If present, this section indicates that the transaction was the initiator, the terminator, or part of an IMS conversation.</td>
</tr>
</tbody>
</table>
### Table 7. Composite record sections in IMS_Vnnn_TRAN (continued)

<table>
<thead>
<tr>
<th>Short description</th>
<th>Name</th>
<th>Prefix</th>
<th>Section contents and explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message queue output</td>
<td>D2</td>
<td>O</td>
<td>X'03' Message queue insert&lt;br&gt;X'35' Message queue enqueue&lt;br&gt;X'31' Message queue GU&lt;br&gt;X'36' Message queue dequeue&lt;br&gt;These records are in sequence for the output message. Note: If the composite record type and subtype is X'FB', only this key section is present in the record. This situation implies that a system-generated message switch has occurred; for example, MTO traffic of broadcast messages. If this section is repeated, it means that multiple outputs were produced from a single processed input message or program. These multiple outputs may have had different destinations.</td>
</tr>
<tr>
<td>Program</td>
<td>PSB</td>
<td>P</td>
<td>X'08' Program scheduled&lt;br&gt;X'07' Program terminated&lt;br&gt;These records are in sequence for the program scheduling that processed the input message. Note: One of these records may be absent when an incomplete IMS log is processed. This situation is especially likely in the case of WFI regions and IFP regions. The X'07' record is the source of program CPU and full function DL/I DB and DC call statistics. Fast Path statistics are found in the X'5937' and X'5938' records. (See Fast Path syncpoint section on page 62 for more information about X'5937' and X'5938' records.) Use caution when analyzing X'07' program termination statistics because they represent only the CPU consumed by a complete scheduling of a program, not necessarily an individual transaction. Program scheduling can process many messages and transactions. Therefore, using it at the lower transaction level is subject to several possible approximating algorithms.</td>
</tr>
</tbody>
</table>
Table 7. Composite record sections in IMS_Vnnn_TRAN (continued)

<table>
<thead>
<tr>
<th>Short description</th>
<th>Name</th>
<th>Prefix</th>
<th>Section contents and explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Msgq transfer/syncpoint</td>
<td>FF</td>
<td>F</td>
<td>X'37' Message commit/transfer X'38' Message failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the commits and transfers of the program processing the input message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> X’37’/X’38’ records may appear for a Fast Path transaction (see EMH Fast Path section on page 63), and X’5937’/X’5938’ records may appear for a full function transaction. If either case occurs, the transaction is known as a <em>mixed mode</em> transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The UOR (unit of recovery) is represented in data terms by the unique 16-byte recovery token field present in all UOR records (and many other associated records). It is unique for the IMS session and represents successful completion of the program.</td>
</tr>
<tr>
<td>Fast Path syncpoint</td>
<td>FP</td>
<td>U</td>
<td>X’5937’ Fast Path commit X’5938’ Fast Path failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the commits and transfers of the program processing the input message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> X’37’/X’38’ records may appear for a Fast Path transaction (see EMH Fast Path section on page 63), and X’5937’/X’5938’ records may appear for a full function transaction. If either case occurs, the transaction is known as a <em>mixed mode</em> transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The X’5937’ and X’5938’ records are the source of Fast Path statistics. For CPU and DL/I DB and DC call statistics, see the X’07’ in the PSB section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The UOR (unit of recovery) is represented in data terms by the unique 16-byte recovery token field present in all UOR records (and many other associated records). It is unique for the IMS session and represents successful completion of the program.</td>
</tr>
<tr>
<td>Short description</td>
<td>Name</td>
<td>Prefix</td>
<td>Section contents and explanation</td>
</tr>
<tr>
<td>----------------------</td>
<td>------</td>
<td>--------</td>
<td>---------------------------------------------------------</td>
</tr>
<tr>
<td>ESS</td>
<td>ESS</td>
<td>X</td>
<td>X'56' ESS connect/call/commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This record is in sequence for the commits and other connections and communications to an IMS external subsystem.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> An IMS external subsystem may be a DBCTL-connected IMS DB system, DB2, or any ESS-connected database system.</td>
</tr>
<tr>
<td>EMH Fast Path</td>
<td>EMH</td>
<td>E</td>
<td>X'5901' EMH input</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5903' EMH output</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5936' EMH dequeue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the stages of logging of an IMS Fast Path EMH-scheduled transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> If this section is present, a type X'5937' or X'5938' record is present in the UOR section.</td>
</tr>
</tbody>
</table>
Log and record definitions

### Composite record types and subtypes in IMS_Vnnn_TRAN

Table 8 shows which record types and subtypes contain which sections:
- **X** The record subtype contains the section.
- **-** The record subtype does not contain the section.
- **** The record subtype may or may not contain the section.

Table 8. Composite record types and subtype sections in IMS_Vnnn_TRAN

<table>
<thead>
<tr>
<th>Composite record type</th>
<th>Description</th>
<th>ROOT</th>
<th>Input (D1)</th>
<th>SPA</th>
<th>Output (D2)</th>
<th>PSB</th>
<th>FF</th>
<th>FP</th>
<th>ESS</th>
<th>EMH</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'xxFF'</td>
<td>Full function transaction or message-driven BMP</td>
<td>**</td>
<td>X</td>
<td>**</td>
<td>X</td>
<td>X</td>
<td>**</td>
<td>-</td>
<td>**</td>
<td>-</td>
</tr>
<tr>
<td>X'xxFE'</td>
<td>Full function transaction or message-driven BMP without output</td>
<td>**</td>
<td>X</td>
<td>**</td>
<td>-</td>
<td>X</td>
<td>**</td>
<td>-</td>
<td>**</td>
<td>-</td>
</tr>
<tr>
<td>X'xxFD'</td>
<td>Non-message-driven BMP with output</td>
<td>-</td>
<td>-</td>
<td>**</td>
<td>X</td>
<td>X</td>
<td>**</td>
<td>-</td>
<td>**</td>
<td>-</td>
</tr>
<tr>
<td>X'xxFC'</td>
<td>BMP or non-message program commits</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>**</td>
<td>-</td>
<td>**</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>X'xxFB'</td>
<td>System-generated message switch</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>X'xxFA'</td>
<td>CNT-generated message switch (includes MSC/ISC/FES)</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>X'xxEF'</td>
<td>EMH mixed mode with program schedule present</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>**</td>
<td>**</td>
<td>X</td>
</tr>
<tr>
<td>X'xxEE'</td>
<td>EMH Fast Path with program schedule present</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>X</td>
</tr>
<tr>
<td>X'xxED'</td>
<td>EMH mixed mode/program schedule unavailable</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>**</td>
<td>**</td>
<td>**</td>
<td>X</td>
</tr>
<tr>
<td>X'xxEC'</td>
<td>EMH Fast Path/program schedule unavailable</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>**</td>
<td>**</td>
<td>X</td>
</tr>
</tbody>
</table>
Chapter 6. IMS performance feature data tables and lookup tables

The Tivoli Decision Support for OS/390 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 OS/390 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 chapter describes the data tables and lookup table used by the IMS Performance feature. It includes an explanation of the naming standard used. For information about the relationships between tables and between records and tables, see “IMS Performance feature data flow” on page 28.

Naming standard for tables

Names of the IMS Performance feature tables use this format:

\[ \text{IMS}_\text{content}_\text{suffix} \]

Where:

- **content** is a description (for example, IMS_USER_TRAN for transactions by user).
- **suffix** indicates the summarization level of the data in the table (for example, IMS_USER_TRAN_H for IMS transactions by user 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, a description of each of the common reference columns, and a description of each of the data columns.

Key columns are marked with a **k**. Common reference columns are marked with an **r** and come after the last key column.

Data columns come after the last common reference column and are sorted in alphabetic order, with any underscores ignored.
Data tables and lookup tables

For each IMS Performance feature subcomponent, the tables appear in alphabetic order, with any underscores and suffixes ignored.

**Note:** Tables with similar contents (that is, tables with the same name but different suffixes) are described under one heading. For example, "IMS_TRANSACTION_H, _D, _W" on page 69 contains information about three similar tables:

- IMS_TRANSACTION_H
- IMS_TRANSACTION_D
- IMS_TRANSACTION_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 Performance feature uses the control tables DAY_OF_WEEK and PERIOD_PLAN, which are used by many Tivoli Decision Support for OS/390 features. For complete descriptions of these control tables, refer to the Administration Guide.

**IMS log records component data tables**

Because the IMS log records component does not update tables, Tivoli Decision Support for OS/390 does not provide tables for this component.

**IMS collect component data tables**

The data tables for the IMS collect component are grouped by subcomponent:

- **Transaction subcomponent tables**
  - IMS_TRANSACTION_H
  - IMS_TRANSACTION_D
  - IMS_TRANSACTION_W
  - IMS_USER_APPL_D
  - IMS_USER_APPL_W
  - IMS_USER_TRAN_H
  - IMS_USER_TRAN_D
  - IMS_USER_TRAN_W
Data tables and lookup tables

- **System subcomponent tables**
  
  IMS_SYSTEM_Q  
  IMS_SYSTEM_D  

- **Application subcomponent tables**
  
  IMS_APPLICATION_H  
  IMS_APPLICATION_W  

- **Statistics subcomponent tables**
  
  IMS_CHKPT_IOSAM_T  
  IMS_CHKPT_POOLS_T  
  IMS_CHKPT_REGION_T  
  IMS_CHKPT_STATS_T  
  IMS_CHKPT_VSAM_T  

The IMS Performance feature maintains separate counts of:

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

The IMS Performance 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, system, and application subcomponents contain information that is:

- Taken from the records produced by the log procedure and record procedure DRL3I
2  
- Supported by record definitions IMS_Vnnn_COMP and IMS_Vnnn_R2 with record procedure definition DRL3I
2  

So, each table contains:

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

All the data tables for the transaction, system, and application subcomponents contain the same common reference columns and data columns, but their key columns vary. Their common reference columns and data columns are described only in “IMS_TRANSACTION_H, _D, _W” on page 69.
Data tables and lookup tables

Table and key column cross-reference

Figure 17 shows an overview of which key columns are present in which IMS collect component data tables.

Table:

<table>
<thead>
<tr>
<th>Table</th>
<th>IMG_APPLICATION_W</th>
<th>IMG_APPLICATION_H</th>
<th>IMG_USER_APPL_W</th>
<th>IMG_USER_APPL_D</th>
<th>IMG_SYSTEM_D</th>
<th>IMG_SYSTEM_H</th>
<th>IMG_TRANSACTION_H</th>
<th>IMG_TRANSACTION_W</th>
<th>IMG_TRANSACTION_D</th>
<th>IMG_USER_TRAN_W</th>
<th>IMG_USER_TRAN_D</th>
<th>IMG_USER_TRAN_H</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>MSG_SYSTEM_ID</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>MSG_TRANSACTION_TYPE</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>REGION_PST_ID</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>USER_ID</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>DATE</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TIME</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

Figure 17. Key columns in transaction, system, and, application subcomponent tables
Transaction subcomponent tables

The data tables described in this section are for the transaction subcomponent. These tables store counts of transactions, resources used, and response times by transaction code and user ID. They are used for performance, capacity, and service-level tuning and troubleshooting.

IMS_TRANSACTION_H, _D, _W

These tables contain hourly, daily, and weekly statistics on counts of transactions, resources used, and response times summarized by transaction name. They contain information that includes data for message-queue-driven transactions and BMPs, nonmessage-driven BMPs, EMH-driven Fast Path transactions, and message switches.

You can use these tables to identify transaction utilization, resource consumption, and subsequent elapse time, transmission, and queuing effects on the IMS system.

Note: The data columns and common reference columns of these tables are also used by the IMS_USER_TRAN_x, IMS_SYSTEM_x, IMS_APPLICATION_x, and IMS_USER_APPL_x tables.

The default retention periods for these tables are:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_TRANSACTION_H</td>
<td>1 day</td>
</tr>
<tr>
<td>IMS_TRANSACTION_D</td>
<td>35 days</td>
</tr>
<tr>
<td>IMS_TRANSACTION_W</td>
<td>365 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activities occurred. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.00.00. This applies only to the _H table.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2invL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td>CHAR(2)</td>
<td>The assigned transaction class.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| TRANSACTION_TYPE   | k CHAR(2) | The transaction type based on (DRL_PRTM + DRL_IQTM) / 10 value ranges. Possible values are:  
|                    |           | **F = FAST** (0.0 - 0.8)  
|                    |           | **G = GOOD** (0.8 - 1.5)  
|                    |           | **M = MEDIUM** (1.5 - 3.0)  
|                    |           | **L = LOW** (3.0 - 10.0)  
|                    |           | **S = SLOW** (above 10)  |
| TRANSACTION_CLASS  | k CHAR(2) | The assigned transaction class.  |
| TRANSACTION_TYPE   | k CHAR(2) | The transaction type based on (DRL_PRTM+DRL_IQTM)/10 value ranges. It could be:  
|                    |           | **F = FAST** (0.0 - 0.8)  
|                    |           | **G = GOOD** (0.8 - 1.5)  
|                    |           | **M = MEDIUM** (1.5 - 3.0)  
|                    |           | **L = LOW** (3.0 - 10.0)  
<p>|                    |           | <strong>S = SLOW</strong> (above 10)  |
| IMS_APPLID         | r CHAR(8) | The VTAM applid for the IMS system. Derived from the runtime parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS applid of a summarization group.  |
| IMS_CTRL_REGION    | r CHAR(8) | The MVS and JES name of the IMS control region address space. This is derived from the runtime parameter :IMS_CTRL_REGION (because few IMS log records contain this field) and selected as the first IMS control region name of a summarization group.  |
| IMS_VERSION        | r CHAR(4) | The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.  |
| APPC_TPI_COMMITS   | FLOAT     | The total number of program commits for the APPC activity. This is derived from the APPC X'37' program commit records. Each commit is the result of DL/I calls to modify databases.  |
| APPC_TPI_CPU_SEC   | FLOAT     | The sum of execution timer units, derived from the APPC TPI termination record (record type X'0A07') divided by 38 400 (the number of timer units that can be accumulated in 1 second of CPU activity).  |
| APPC_TPI_DPSBCALLS | FLOAT     | The total number of APPC TPI detach PSB (DPSBs), derived from the number of type X'07' records. Record type X'07' is the same as that used for full function DL/I PSB resource statistics, but is written each time the TPI makes a DPSB call to deallocate the PSB used for DL/I access.  |
| APPC_TPI_ENDED     | FLOAT     | The total number of APPC TPI termination records (record type X'0A07'). This represents the number of APPC transaction program instance terminations.  |
| APPC_TPI_STARTED   | FLOAT     | The total number of APPC TPI schedule records (record type X'0A08'). This represents the number of APPC transaction program instance schedules.  |
| APPC_NETID         | CHAR(8)   | The destination network ID for the APPC session.  |</p>
<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPC_Mode_Name</td>
<td>CHAR(8)</td>
<td>The mode name for the APPC session.</td>
</tr>
<tr>
<td>APPC_TIBQAB_T</td>
<td>CHAR(4)</td>
<td>The TIB/QAB origin or destination type for the APPC session.</td>
</tr>
<tr>
<td>APPC_TIBQAB_A</td>
<td>CHAR(4)</td>
<td>The TIB/QAB origin or destination address for the APPC session.</td>
</tr>
<tr>
<td>APPC_NetID</td>
<td>CHAR(8)</td>
<td>The destination Network ID for the APPC session</td>
</tr>
<tr>
<td>BMP_DS_LMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the long message queue for nonmessage-driven BMPs.</td>
</tr>
<tr>
<td>BMP_DS_LMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of messages inserted to the long message queue for nonmessage-driven BMPs.</td>
</tr>
<tr>
<td>BMP_DS_SMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the short message queue for nonmessage-driven BMPs.</td>
</tr>
<tr>
<td>BMP_DS_SMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of messages inserted to the short message queue for nonmessage-driven BMPs.</td>
</tr>
<tr>
<td>BMP_FF_ABORTS</td>
<td>FLOAT</td>
<td>The total number of nonmessage-driven BMP full function commits aborted.</td>
</tr>
<tr>
<td>BMP_FF_COMMITS</td>
<td>FLOAT</td>
<td>The total number of nonmessage-driven BMP full function commits completed.</td>
</tr>
<tr>
<td>BMP_FP_ABORTS</td>
<td>FLOAT</td>
<td>The total number of nonmessage-driven BMP Fast Path commits aborted.</td>
</tr>
<tr>
<td>BMP_FP_COMMITS</td>
<td>FLOAT</td>
<td>The total number of nonmessage-driven BMP Fast Path commits completed.</td>
</tr>
<tr>
<td>BMP_OUTPUT_MSGS</td>
<td>FLOAT</td>
<td>The total number of message queue output messages issued by nonmessage-driven BMPs.</td>
</tr>
<tr>
<td>BMP_PROCESS_SEC</td>
<td>FLOAT</td>
<td>The total processing time of nonmessage-driven BMP programs, in seconds.</td>
</tr>
<tr>
<td>BMP_PROGRAMS</td>
<td>FLOAT</td>
<td>The total number of nonmessage-driven BMP programs executed.</td>
</tr>
<tr>
<td>DL1_Cmd_Calls</td>
<td>FLOAT</td>
<td>The total number of DL/I CMD calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I CMD calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DL1DB_Calls</td>
<td>FLOAT</td>
<td>The total number of DL/I database calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate total number of DL/I database calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DL1DBCTL_DBIOS</td>
<td>FLOAT</td>
<td>The total number of DB I/Os for DBCTL, derived from the count stored in the program termination record (record type X'07'). This represents the approximate total number of DB I/Os for DBCTL that the program issued while the transactions were active.</td>
</tr>
</tbody>
</table>
## Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLIDBCTL_DBIO_SEC</td>
<td>FLOAT</td>
<td>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'). This represents the approximate elapsed time of DB I/Os for DBCTL that the program undertook while the transactions were active.</td>
</tr>
<tr>
<td>DLIDBCTL_LOCK_SEC</td>
<td>FLOAT</td>
<td>The total elapsed time for locking for DBCTL, in seconds, derived from the value stored in the program termination record (record type X'07'). This represents the approximate elapsed time of locking for DBCTL that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_DLET_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database DLET calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GHN_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GHN calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GHN calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GHNP_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GHNP calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GHNP calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GHU_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GHU calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GHU calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GN_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GN calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GN calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GNP_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GNP calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GNP calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_GU_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database GU calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database GU calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDB_ISRT_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database ISRT calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database ISRT calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DLIDB_REPL_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I database REPL calls issued, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I database REPL calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDC_GN_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I message queue GN calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I message queue GN calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDC_GU_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I message queue GU calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I message queue GU calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDC_ISRT_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I message queue ISRT calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I message queue ISRT calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLIDC_PURGE_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I message queue purge calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I message queue purge calls that the program issued while the transactions were active.</td>
</tr>
</tbody>
</table>
| DLI_EXCL_DEQUEUES       | FLOAT     | The total number of DL/I-exclusive dequeue calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of exclusive dequeue calls that the program issued while the transactions were active.  
This column indicates the degree of serialization and program isolation during this interval, either throughout the system or resulting from this user or transaction. |
| DLI_EXCL_ENQUEUES       | FLOAT     | The total number of DL/I-exclusive enqueue calls, derived from the count stored in the program termination record (record type X'07'). This represents the sum of the approximate number of exclusive enqueue calls that the program issued while the transactions were active.  
This column indicates the degree of serialization and program isolation during this interval, either throughout the system or resulting from this user or transaction. |
| DLI_EXCL_ENQWAITS       | FLOAT     | The total number of waits for DL/I-exclusive enqueue calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of waits for exclusive enqueue calls that the program issued while the transactions were active.  
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. |
### Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLI_GCMD_CALLS</td>
<td>FLOAT</td>
<td>The total number of DL/I GCMD calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I GCMD calls that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_QCMD_DEQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I queue command dequeue calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I queue command dequeue calls that occurred while the transactions were active.</td>
</tr>
<tr>
<td>DLI_QCMD_ENQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I queue command enqueue calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I queue command enqueue calls that occurred while the transactions were active.</td>
</tr>
<tr>
<td>DLI_QCMD_ENQWAITS</td>
<td>FLOAT</td>
<td>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'). This represents the approximate number of waits for DL/I queue commands and enqueues that occurred while the transactions were active.</td>
</tr>
<tr>
<td>DLISQ6_ACCUM_SEC</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_TEST_DEQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I test dequeues, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I test dequeues that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_TEST_ENQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I test enqueues, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I test enqueues that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_TEST_ENQWAITS</td>
<td>FLOAT</td>
<td>The total number of DL/I waits on test enqueues, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I waits on test enqueues that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_UPDT_DEQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I update dequeues, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I update dequeues that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_UPDT_ENQUEUES</td>
<td>FLOAT</td>
<td>The total number of DL/I update enqueues, derived from the count stored in the program termination record (record type X'07'). This represents the approximate number of DL/I update enqueues that the program issued while the transactions were active.</td>
</tr>
<tr>
<td>DLI_UPDT_ENQWAITS</td>
<td>FLOAT</td>
<td>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'). This represents the approximate number of DL/I waits on update and enqueues that the program issued while the transactions were active.</td>
</tr>
</tbody>
</table>
## Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMH_BUFFER_BYTES</td>
<td>FLOAT</td>
<td>The total number of bytes of data written to the EMH (input) and ESRT (output) buffers for input and output messages for Fast Path EMH-driven transactions.</td>
</tr>
<tr>
<td>EMH_DS_LMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the long message queue for EMH-driven Fast Path transactions.</td>
</tr>
<tr>
<td>EMH_DS_LMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of messages inserted to the long message queue for EMH-driven Fast Path transactions.</td>
</tr>
<tr>
<td>EMH_DS_SMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the short message queue for EMH-driven Fast Path transactions.</td>
</tr>
<tr>
<td>EMH_DS_SMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of messages inserted to the short message queue for EMH-driven Fast Path transactions.</td>
</tr>
<tr>
<td>EMH_FF_ABORTS</td>
<td>FLOAT</td>
<td>The total number of mixed mode EMH-initiated transactions that aborted their full function commits.</td>
</tr>
<tr>
<td>EMH_FF_COMMITS</td>
<td>FLOAT</td>
<td>The total number of mixed mode EMH-initiated transactions that completed their full function commits.</td>
</tr>
<tr>
<td>EMH_FP_ABORTS</td>
<td>FLOAT</td>
<td>The total number of EMH-initiated transactions that aborted their Fast Path commits.</td>
</tr>
<tr>
<td>EMH_FP_COMMITS</td>
<td>FLOAT</td>
<td>The total number of EMH-initiated transactions that completed their Fast Path commits.</td>
</tr>
<tr>
<td>EMH_INPUT_SEC</td>
<td>FLOAT</td>
<td>The total time spent waiting in the input EMH buffer for the balancing group, in seconds.</td>
</tr>
<tr>
<td>EMH_MSGQ_OUTPUTS</td>
<td>FLOAT</td>
<td>The total number of MSGQ output messages issued by EMH-initiated transactions.</td>
</tr>
<tr>
<td>EMH_NETWORK_SEC</td>
<td>FLOAT</td>
<td>The total time 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.</td>
</tr>
<tr>
<td>EMH_OUTPUT_SEC</td>
<td>FLOAT</td>
<td>The total time spent in the IMS output ESRT buffer waiting for transmission to the ultimate destination, in seconds.</td>
</tr>
<tr>
<td>EMH_PROCESS_SEC</td>
<td>FLOAT</td>
<td>The total elapsed time that EMH-driven Fast Path transactions spent processing in the dependent regions, in seconds.</td>
</tr>
<tr>
<td>EMH_RESPONSE_SEC</td>
<td>FLOAT</td>
<td>The sum of the total end-to-end user perceived response time, in seconds. This normally includes user think time and therefore cannot be used easily to gauge true end-user response times.</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_1</td>
<td>FLOAT</td>
<td>The total number of IMS Fast Path transactions whose transit time was less than the user-specified boundary 1 (default for boundary is 1 second).</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_2</td>
<td>FLOAT</td>
<td>The total number of IMS Fast Path transactions whose transit time was less than the user-specified boundary 2 (default for boundary is 2 seconds).</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_3</td>
<td>FLOAT</td>
<td>The total number of IMS Fast Path transactions whose transit time was less than the user-specified boundary 3 (default for boundary is 5 seconds).</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_4</td>
<td>FLOAT</td>
<td>The total number of IMS Fast Path transactions whose transit time was less than the user-specified boundary 4 (default for boundary is 10 seconds).</td>
</tr>
<tr>
<td>EMH_TRANSACTIONS</td>
<td>FLOAT</td>
<td>The total number of IMS Fast Path transactions for the given interval and unique key combination.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EMH_TRANSIT_SEC</td>
<td>FLOAT</td>
<td>The total time spent in the IMS system from input to output, excluding the network transmission time, in seconds.</td>
</tr>
<tr>
<td>FP_CI_HNH_CONT</td>
<td>FLOAT</td>
<td>The total number of Fast Path CI contentions between 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 HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>FP_CI_NHNH_CONT</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_COMBINATIONS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_DEDB_BFR_WAITS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_DEDB_CALLS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_DEDB_HSSP_CALLS</td>
<td>FLOAT</td>
<td>The total number of Fast Path DEDB calls by HSSP, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of DEDB calls by HSSP that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>FP_DEDB_HSSP_PUTS</td>
<td>FLOAT</td>
<td>The total number of Fast Path DEDB PUTs by HSSP on image copy 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 by HSSP on image copy data sets that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>FP_DEDB_NH_PUTGETS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_LOGGED_CI</td>
<td>FLOAT</td>
<td>The total number of Fast Path whole CIs 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.</td>
</tr>
<tr>
<td>FP_MSDB_CALLS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FP_OVERFLOW_BFR</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>FP_UOW_HNH_CONT</td>
<td>FLOAT</td>
<td>The total number of Fast Path UOW contentions between 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 UOW contentions between HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>FP_UOW_NHNH_CONT</td>
<td>FLOAT</td>
<td>The total number of Fast Path UOW 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 UOW contentions between non-HSSP and non-HSSP EPCBs that the program experienced while the transactions were active.</td>
</tr>
<tr>
<td>MSGQ_CONV_STARTED</td>
<td>FLOAT</td>
<td>The total number of conversations started during the given interval and unique key combination (for example, user and transaction code) as indicated by the presence of a type X'11' conversation started record.</td>
</tr>
<tr>
<td>MSGQ_CONV_TRAN</td>
<td>FLOAT</td>
<td>The total number of IMS conversational transactions for the given interval and unique key combination (for example, user and transaction code).</td>
</tr>
<tr>
<td>MSGQ_DS_LMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the long message queue for message-driven transactions, BMP programs, and message switches.</td>
</tr>
<tr>
<td>MSGQ_DS_LMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of segments inserted to the long message queue for message-driven transactions, BMP programs, and message switches (that is all input segments and all output segments destined to CNT, but no output segments destined to SMB).</td>
</tr>
<tr>
<td>MSGQ_DS_SMSG_BYTES</td>
<td>FLOAT</td>
<td>The total number of text bytes inserted to the short message queue for message-driven transactions, BMP programs, and message switches.</td>
</tr>
<tr>
<td>MSGQ_DS_SMSG_ISRT</td>
<td>FLOAT</td>
<td>The total number of segments inserted to the short message queue for message-driven transactions, BMP programs, and message switches. (that is all input segments and all output segments destined to CNT, but no output segments destined to SMB).</td>
</tr>
<tr>
<td>MSGQ_DS_SPA_ISRT</td>
<td>FLOAT</td>
<td>The total number of messages inserted to the scratchpad area.</td>
</tr>
<tr>
<td>MSGQ_FF_ABORTS</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs that aborted their full function commits.</td>
</tr>
<tr>
<td>MSGQ_FF_COMMITS</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs that completed their full function commits.</td>
</tr>
<tr>
<td>MSGQ_FP_ABORTS</td>
<td>FLOAT</td>
<td>The total number of mixed-mode, message-driven transactions and BMP programs that aborted their Fast Path commits.</td>
</tr>
<tr>
<td>MSGQ_FP_COMMITS</td>
<td>FLOAT</td>
<td>The total number of mixed-mode MPP and BMP programs that completed their Fast Path commits.</td>
</tr>
<tr>
<td>MSGQ_INPUT_MSGS</td>
<td>FLOAT</td>
<td>The total number of input messages received from the data communication system. This number does not include transactions generated by program-to-program switches.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MSGQ_INPUT_SEC</td>
<td>FLOAT</td>
<td>The total time, in seconds, that message-driven transactions and BMP programs spent on the IMS input message queue, including input queue time for program-to-program switch transactions.</td>
</tr>
<tr>
<td>MSGQ_MSG_SWITCHES</td>
<td>FLOAT</td>
<td>The total number of message switches, including MSC and ISC messages.</td>
</tr>
<tr>
<td>MSGQ_NETWORK_SEC</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>MSGQ_OUTPUT_MSGS</td>
<td>FLOAT</td>
<td>The total number of output messages issued by message-driven transactions and BMP programs.</td>
</tr>
<tr>
<td>MSGQ_OUTPUT_SEC</td>
<td>FLOAT</td>
<td>The total time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.</td>
</tr>
<tr>
<td>MSGQ_PROCESS_SEC</td>
<td>FLOAT</td>
<td>The total elapsed time that message-driven transactions and BMP programs spent processing in the dependent regions, in seconds.</td>
</tr>
<tr>
<td>MSGQ_RESPONDENCES</td>
<td>FLOAT</td>
<td>The total number of responding message-driven transactions and BMP programs that sent messages to the originating terminal.</td>
</tr>
<tr>
<td>MSGQ_RESPONSE_SEC</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_1</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs when the transit time was less than the user-specified boundary 1 (default for boundary is 1 second).</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_2</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs when the transit time was less than the user-specified boundary 2 (default for boundary is 2 seconds).</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_3</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs when the transit time was less than the user-specified boundary 3 (default for boundary is 5 seconds).</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_4</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs when the transit time was less than the user-specified boundary 4 (default for boundary is 10 seconds).</td>
</tr>
<tr>
<td>MSGQ_TRANSACTIONS</td>
<td>FLOAT</td>
<td>The total number of message-driven transactions and BMP programs.</td>
</tr>
<tr>
<td>MSGQ_TRANSIT_SEC</td>
<td>FLOAT</td>
<td>The total time, in seconds, message-driven 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.</td>
</tr>
<tr>
<td>PGM_CPU_SEC</td>
<td>FLOAT</td>
<td>The total dependent region CPU TCB 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). This column represents the sum of the approximate number of CPU seconds of program execution time while the transactions were active.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PGM_FAILURES</td>
<td>FLOAT</td>
<td>The total number of program abends.</td>
</tr>
<tr>
<td>PGM_TIMER_UNITS</td>
<td>FLOAT</td>
<td>The total number of dependent region CPU timer units, derived from the count of CPU timer units stored in the program termination record (record type X'07'). This column represents the sum of the approximate number of CPU seconds of program execution time while the transactions were active.</td>
</tr>
<tr>
<td>RESPONSE_BNDY1_SEC</td>
<td>FLOAT</td>
<td>The first transaction transit-time boundary, in seconds (default is 1 second). This is used to determine the number of transactions within this boundary.</td>
</tr>
<tr>
<td>RESPONSE_BNDY2_SEC</td>
<td>FLOAT</td>
<td>The second transaction transit-time boundary, in seconds (default is 2 seconds). This is used to determine the number of transactions within this boundary.</td>
</tr>
<tr>
<td>RESPONSE_BNDY3_SEC</td>
<td>FLOAT</td>
<td>The third transaction transit-time boundary, in seconds (default is 5 seconds). This is used to determine the number of transactions within this boundary.</td>
</tr>
<tr>
<td>RESPONSE_BNDY4_SEC</td>
<td>FLOAT</td>
<td>The fourth transaction transit-time boundary, in seconds (default is 10 seconds). This is used to determine the number of transactions within this boundary.</td>
</tr>
</tbody>
</table>
These tables contain hourly, daily, and weekly statistics on counts of transactions, resources used, and response times summarized by transaction name and user ID. They contain information that includes data for message-queue-driven transactions and BMPs, nonmessage-driven BMPs, EMH-driven Fast Path transactions, and message switches.

You can use these tables to identify what users did, how their volumes differed, and their response-time experiences.

The default retention periods for these tables are:
- IMS_USER_TRAN_H 1 day
- IMS_USER_TRAN_D 8 days
- IMS_USER_TRAN_W 35 days

**Note:** Only the key columns of these tables are described here. These tables also contain all the common reference columns and data columns described in “IMS_TRANSACTION_H, _D, _W” on page 69.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH:00:00. This applies only to the _H table.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2INnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>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 and RACF utilities and products.</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td>CHAR(8)</td>
<td>The IMS-defined logical name for the terminal used to request the transaction. This column is associated with a physical terminal recognized by the network.</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td>CHAR(8)</td>
<td>The network-recognized name for the terminal used to request the transaction. IMS maintains a relationship between this name, which is known to the network, and its own logical terminal name.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction the user requested. For full function activity, this column is the name of the scheduler message block (SMB). For Fast Path activity, this column is the routing code. For APPC activity, this column is the transaction program instance (TPI).</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
</tbody>
</table>
## IMS_TRAN_TYPE key column

The IMS_TRAN_TYPE key column, which is used in the IMS_USER_TRAN_H table, contains transaction characteristics. It also appears in the R1 report and in the composite record produced by the R2 record procedure. See “Record procedures” on page 19 for information about the R1 and R2 record procedures.

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

The first byte is the transaction-type flag:
- **F** A Fast Path program processed the transaction.
- **M** An MPP processed the transaction.
- **B** A BMP program was processed or processed the message.
- **S** A message switch has been processed by IMS.
- **T** An MTO or IMS generated message switch has been processed by IMS.
- **U** An output message that was generated by a transaction has been found but could not be connected to the transaction (an unconnected output).
- **-** The activity could not be identified.

The second byte is the region type flag:
- **W** Wait for input processing was undertaken for this transaction.

The third byte is the conversation flag:
- **S** A conversation was started.
- **C** A conversation continues.
- **E** A conversation was ended.

The fourth byte is the program-to-program switch flag:
- **P** A program-to-program switch was generated from this parent transaction.
- **C** A program-to-program switch generated this transaction, and it generated another program-to-program switch.
- **E** A program-to-program switch generated this transaction, and it did not generate any program-to-program switches.

The fifth byte is the MSC/ISC flag:
### Data tables and lookup tables

| M  | The transaction or message switch was part of an MSC sequence of messages and transactions involving more than one IMS system. |
| I  | The transaction or message switch was part of an ISC sequence of messages and transactions. |

The sixth byte is the mixed mode flag:

| X  | A full function transaction accessed Fast Path databases or a Fast Path transaction accessed full function databases. |
| Q  | A Fast Path transaction issued message output. |

The seventh byte is the transaction completion status flag:

| R  | A transaction aborted and was retried by IMS. |
| J  | A transaction aborted and the input message was rejected by IMS, preventing further retry. |
| C  | An input message was canceled by IMS. |

The eighth byte is the transaction abend status flag:

| A  | A transaction aborted or a program abended. |
System subcomponent tables

The data tables described in this section are for the system subcomponent. These tables store counts of transactions, resources used, and response times by IMS system. They provide a view of IMS capacity and service-level trend analysis that allows service-level monitoring by system, and trend analysis of volumes and response times.

IMS_SYSTEM_Q, IMS_SYSTEM_D

These tables contain quarter-hourly and daily statistics on counts of transactions, resources used, and response times summarized by IMS system. They contain information that includes data for message-queue-driven transactions and BMPs, nonmessage-driven 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.

The default retention periods for these tables are:

<table>
<thead>
<tr>
<th>Table</th>
<th>Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_SYSTEM_Q</td>
<td>70 days</td>
</tr>
<tr>
<td>IMS_SYSTEM_D</td>
<td>1095 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH:15:00. This applies only to the _Q table.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2ImnnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>REGION_PST_ID</td>
<td>CHAR(2)</td>
<td>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.</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td>CHAR(2)</td>
<td>The assigned transaction class.</td>
</tr>
</tbody>
</table>
Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_TYPE</td>
<td>CHAR(2)</td>
<td>The transaction type based on ((\text{DRL_PRTM} + \text{DRL_IQT}) / 10) value ranges. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(F = \text{FAST} \ (0.0 - 0.8))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(G = \text{GOOD} \ (0.8 - 1.5))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(M = \text{MEDIUM} \ (1.5 - 3.0))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(L = \text{LOW} \ (3.0 - 10.0))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(S = \text{SLOW} \ (\text{above 10}))</td>
</tr>
</tbody>
</table>

| TRANSACTION_TYPE  | CHAR(2)   | The transaction type based on \((\text{DRL\_PRTM}+\text{DRL\_IQT})/10\) value ranges. It could be:          |
|                   |           | \(F = \text{FAST} \ (0.0 - 0.8)\)                                           |
|                   |           | \(G = \text{GOOD} \ (0.8 - 1.5)\)                                           |
|                   |           | \(M = \text{MEDIUM} \ (1.5 - 3.0)\)                                         |
|                   |           | \(L = \text{LOW} \ (3.0 - 10.0)\)                                          |
|                   |           | \(S = \text{SLOW} \ (\text{above 10})\)                                     |

| PERIOD_NAME       | 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. |

**Note:** Only the key columns of these tables are described here. These tables also contain all the common reference columns and data columns described in "IMS\_TRANSACTION\_H, \_D, \_W" on page 69.
The data tables described in this section are for the application subcomponent. These tables store counts of transactions, resources used, and response times by application code and transaction code. They are oriented toward a performance and capacity view of IMS transaction activity.

**IMS_APPLICATION_H, _W**

These tables contain hourly and weekly statistics on counts of transactions, resources used, and response times summarized by application. They contain information that includes data for message-queue-driven transactions and BMPs, nonmessage-driven BMPs, EMH-driven Fast Path transactions, and message switches.

You can use these tables for service-level monitoring by application and for trend analysis of volumes and response times.

This table is updated by the IMSAPPLICATION lookup table.

The default retention periods for these tables are:
- IMS_APPLICATION_H: 70 days
- IMS_APPLICATION_W: 1095 days

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activities occurred. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.00.00. This applies only to the _H table.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2Imml supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td>CHAR(18)</td>
<td>The name of the business application responsible for the transaction processing activity. This column is derived from the IMS_APPLICATION table using the lookup function with the MVS_SYSTEM_ID, TRANSACTION_NAME, and PROGRAM_NAME columns as reference.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
</tbody>
</table>

Note: Only the key columns of these tables are described here. These tables also contain all the common reference columns and data columns described in “IMS_TRANSACTION_H, _D, _W” on page 69.
Data tables and lookup tables

**IMS_USER_APPL_D, _W**

These tables contain daily and weekly statistics on counts of transactions, resources used, and response times summarized by application name and user ID. They contain information that includes data for message-queue-driven transactions and BMPs, nonmessage-driven BMPs, EMH-driven Fast Path transactions, and message switches.

You can use these tables for short trend analysis, service-level monitoring by user and application, and for trend analysis of volumes and resource consumption.

This table is updated by the IMS_APPLICATION lookup table.

The default retention periods for these tables are:

- IMS_USER_APPL_D: 35 days
- IMS_USER_APPL_W: 365 days

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activities occurred. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter _MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2InnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>USER_ID</td>
<td>CHAR(8)</td>
<td>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 and RACF utilities and products.</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td>CHAR(18)</td>
<td>The name of the business application responsible for the transaction processing activity. This column is derived from the IMS_APPLICATION table using the lookup function with the MVS_SYSTEM_ID, TRANSACTION_NAME, and PROGRAM_NAME columns as reference.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Note:** Only the key columns of these tables are described here. These tables also contain all the common reference columns and data columns described in "IMS_TRANSACTION_H, _D, _W" on page 69.
Statistics subcomponent tables

The data tables described in this section are for the statistics subcomponent. These tables contain information taken from the modified checkpoint and IMS statistics records produced at IMS checkpoint time and mapped by record definitions IMS_Vnnn_STxxxx.

These tables let you analyze IMS buffers, pools, storage, and data set utilization. As with all high-volume, high-performance transaction systems, the correct analysis and tuning of internal buffers and pools is essential for optimum throughput and performance.

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.

The default retention period for this table is 7 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>k</td>
<td>DATE The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k</td>
<td>TIME The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k</td>
<td>TIMESTAMP The full microsecond precision date and timestamp of the time the checkpoint occurred.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(4) The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(8) The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2InnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>BUFFER_SIZE</td>
<td>k</td>
<td>CHAR(8) The size of buffers in this buffer pool.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k</td>
<td>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.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>r</td>
<td>CHAR(8) The VTAM APPLID for the IMS system. This is the ID by which VTAM communicates with the IMS system. It is derived from the run time parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS APPLID of a summarization group.</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>r</td>
<td>SMALLINT The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r</td>
<td>CHAR(8) The MVS and JES name of the IMS control region address space. This is derived from the run time parameter :IMS_CTRL_REGION (because few IM log records contain this field) and selected as the first IMS control region name of a summarization group.</td>
</tr>
</tbody>
</table>
### Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_VERSION</td>
<td>r CHAR(4)</td>
<td>The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.</td>
</tr>
<tr>
<td>BLOCKS_WRITE_PURGE</td>
<td>INTEGER</td>
<td>The number of blocks written by purge requests for the IMS session.</td>
</tr>
<tr>
<td>BUFFER_ALTER_CALLS</td>
<td>INTEGER</td>
<td>The number of buffer alterations for the IMS session.</td>
</tr>
<tr>
<td>BUFFERS</td>
<td>INTEGER</td>
<td>The number of buffers in this buffer pool for the IMS session.</td>
</tr>
<tr>
<td>BUFFERS_LOCKED</td>
<td>INTEGER</td>
<td>The largest number of locked buffers for the IMS session.</td>
</tr>
<tr>
<td>BUFFERS_SEARCHED</td>
<td>FLOAT</td>
<td>The number of buffers searched for the IMS session.</td>
</tr>
<tr>
<td>LOCATE_CALLS</td>
<td>FLOAT</td>
<td>The total number of block requests (LOCATE BYTE, BLOCK, BYTALT) for the IMS session.</td>
</tr>
<tr>
<td>LOCATE_CALLS_SUCC</td>
<td>FLOAT</td>
<td>The number of requests satisfied from pool (I/O not required) for the IMS session.</td>
</tr>
<tr>
<td>LOC_CALLS_WAIT_ID</td>
<td>INTEGER</td>
<td>The number of locate calls that waited for busy ids for the IMS session.</td>
</tr>
<tr>
<td>LOC_CALLS_WAIT_RD</td>
<td>INTEGER</td>
<td>The number of locate call waits because buffer steal was busy reading for the IMS session.</td>
</tr>
<tr>
<td>LOC_CALLS_WAIT_WR</td>
<td>INTEGER</td>
<td>The number of locate call waits because buffer steal was busy writing for the IMS session.</td>
</tr>
<tr>
<td>NEW_BLOCK_CREATES</td>
<td>INTEGER</td>
<td>The number of requests to create a new block or a logical record for the IMS session.</td>
</tr>
<tr>
<td>PERM_WRITE_ERRORS</td>
<td>INTEGER</td>
<td>The number of permanent write error buffers currently locked in the pool for the IMS session.</td>
</tr>
<tr>
<td>POOL_ID</td>
<td>CHAR(4)</td>
<td>The name of the buffer pool for the gathered statistics.</td>
</tr>
<tr>
<td>PURGE_CALLS</td>
<td>INTEGER</td>
<td>The number of purge user requests for the IMS session.</td>
</tr>
<tr>
<td>READ_IO_COUNT</td>
<td>INTEGER</td>
<td>The number of read I/Os (except BISAM RKU requests) for the IMS session.</td>
</tr>
<tr>
<td>WAITS_NO_BUFFER</td>
<td>INTEGER</td>
<td>The number of buffer steal requests that had to wait because no buffers were available for the IMS session.</td>
</tr>
<tr>
<td>WAITS_RLSE_OWN</td>
<td>INTEGER</td>
<td>The number of buffer steal or purge queue requests that had to wait for release ownership requests for the IMS session.</td>
</tr>
<tr>
<td>WRITES_BFR_STEAL</td>
<td>INTEGER</td>
<td>The number of OSAM writes issued (single block writes because of buffer steal) for the IMS session.</td>
</tr>
</tbody>
</table>
**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.

The default retention period for this table is 7 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>The full microsecond precision date and timestamp of the time the checkpoint occurred.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID.</td>
</tr>
<tr>
<td>POOL_ID</td>
<td>CHAR(8)</td>
<td>The name of the buffer pool for the gathered statistics.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>CHAR(8)</td>
<td>The VTAM APPLID for the IMS system. This is the ID by which VTAM communicates with the IMS system. It is derived from the run time parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS APPLID of a summarization group.</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>CHAR(8)</td>
<td>The MVS and JES name of the IMS control region address space. This is derived from the run time parameter :IMS_CTRL_REGION (because few IM log records contain this field) and selected as the first IMS control region name of a summarization group.</td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>CHAR(4)</td>
<td>The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.</td>
</tr>
<tr>
<td>BUFFERS_CURRENT</td>
<td>INTEGER</td>
<td>The current amount of storage allocated since the last checkpoint.</td>
</tr>
<tr>
<td>BUFFERS_HIGH</td>
<td>INTEGER</td>
<td>The maximum storage allocated from this pool since the last checkpoint.</td>
</tr>
<tr>
<td>BUFFER_SIZE</td>
<td>INTEGER</td>
<td>The size of the buffer pool for the gathered statistics.</td>
</tr>
<tr>
<td>BUFFERS_Oversize</td>
<td>INTEGER</td>
<td>The current number of bytes in oversize blocks.</td>
</tr>
</tbody>
</table>
## 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.

The default retention period for this table is 7 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH:MM:SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>The full microsecond precision date and timestamp of the time the checkpoint occurred.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2ImnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>REGION_PST_ID</td>
<td>CHAR(4)</td>
<td>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.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>CHAR(8)</td>
<td>The VTAM APPLID for the IMS system. This is the ID by which VTAM communicates with the IMS system. It is derived from the run time parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS applid of a summarization group.</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>CHAR(8)</td>
<td>The MVS and JES name of the IMS control region address space. This is derived from the run time parameter :IMS_CTRL_REGION (because few IM log records contain this field) and selected as the first IMS control region name of a summarization group.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>r</td>
<td>CHAR(4)  The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS application program active in the region at IMS checkpoint time. For full function activity and Fast Path activity, this column contains the program specification block (PSB) if available. For APPC activity, this column contains the TPI used.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction active in the region at IMS checkpoint time. For full function activity, this column is the name of the scheduler message block (SMB). For Fast Path activity, this column is the routing code. For APPC activity, this column is the transaction program instance (TPI).</td>
</tr>
</tbody>
</table>
### 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.

The default retention period for this table is 7 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>The full microsecond precision date and timestamp of the time the checkpoint occurred.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>CHAR(8)</td>
<td>The VTAM APPLID for the IMS system. This is the ID by which VTAM communicates with the IMS system. It is derived from the run time parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS applid of a summarization group.</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>CHAR(8)</td>
<td>The MVS and JES name of the IMS control region address space. This is derived from the run time parameter :IMS_CTRL_REGION (because few IM log records contain this field) and selected as the first IMS control region name of a summarization group.</td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>CHAR(4)</td>
<td>The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.</td>
</tr>
<tr>
<td>CHKPT_ADDR_SPC_ID</td>
<td>CHAR(8)</td>
<td>The address space ID (ASID) of the active system.</td>
</tr>
<tr>
<td>CHKPT_CPU_ID</td>
<td>CHAR(16)</td>
<td>The CPU ID (serial number) on which the IMS system ran.</td>
</tr>
<tr>
<td>CHKPT_CTRL_TCB</td>
<td>INTEGER</td>
<td>The CTL TCB task time derived from the timer units stored at checkpoint time in the X’4001’ record</td>
</tr>
<tr>
<td>CHKPT_DLI_TCB</td>
<td>INTEGER</td>
<td>The DL/I SAS region TCB (if LSO=S) derived from the timer units stored at checkpoint time in the X’4001’ record.</td>
</tr>
<tr>
<td>CHKPT_HOT_STANDBY</td>
<td>CHAR(8)</td>
<td>The VTAM generic name for the hot standby system.</td>
</tr>
<tr>
<td>CHKPT_IRLM_NAME</td>
<td>CHAR(4)</td>
<td>The IRLM subsystem name.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CHKPT_LMSG_RECORDS</td>
<td>INTEGER</td>
<td>The number of records allocated in the large message queue data set.</td>
</tr>
<tr>
<td>CHKPT_QBLK_RECORDS</td>
<td>INTEGER</td>
<td>The number of records allocated in the queue blocks data set.</td>
</tr>
<tr>
<td>CHKPT_SMSG_RECORDS</td>
<td>INTEGER</td>
<td>The number of records allocated in the short message queue data set.</td>
</tr>
<tr>
<td>FBP_DIR_IO_COUNT</td>
<td>INTEGER</td>
<td>The number of directory I/Os.</td>
</tr>
<tr>
<td>FBP_DIR_NO_ENTRIES</td>
<td>INTEGER</td>
<td>The number of times that there were no directory entries for a block.</td>
</tr>
<tr>
<td>FBP_DIR_REQUESTS</td>
<td>INTEGER</td>
<td>The number of requests satisfied by the directory.</td>
</tr>
<tr>
<td>FBP_FBLOCK_IFETCHQ</td>
<td>INTEGER</td>
<td>The number of format blocks on the immediate fetch queue.</td>
</tr>
<tr>
<td>FBP_FBLOCK_IGNORES</td>
<td>INTEGER</td>
<td>The number of format blocks ignored.</td>
</tr>
<tr>
<td>FBP_FBLOCK_REQ</td>
<td>INTEGER</td>
<td>The number of format block requests.</td>
</tr>
<tr>
<td>FBP_FBLOCK_WASHES</td>
<td>INTEGER</td>
<td>The number of format block washes.</td>
</tr>
<tr>
<td>FBP_IFETCH_FBLOCKQ</td>
<td>INTEGER</td>
<td>The number of immediate fetches on the format blocks queue.</td>
</tr>
<tr>
<td>FBP_IFETCH_IFETCHQ</td>
<td>INTEGER</td>
<td>The number of immediate fetches on the immediate fetch queue.</td>
</tr>
<tr>
<td>FBP_IFETCH_IO_CNT</td>
<td>INTEGER</td>
<td>The number of immediate fetch I/Os.</td>
</tr>
<tr>
<td>FBP_IFETCH_PFETCHQ</td>
<td>INTEGER</td>
<td>The number of immediate fetches on the prefetch queue.</td>
</tr>
<tr>
<td>FBP_IFETCH_REQ</td>
<td>INTEGER</td>
<td>The number of immediate fetch requests.</td>
</tr>
<tr>
<td>FBP_IO_ERRORS</td>
<td>INTEGER</td>
<td>The number of I/O errors (point or read macro).</td>
</tr>
<tr>
<td>FBP_IO_WAITS</td>
<td>INTEGER</td>
<td>The number of I/O request waits.</td>
</tr>
<tr>
<td>FBP_PFETCH_FBLOCKQ</td>
<td>INTEGER</td>
<td>The number of prefetches on the format blocks queue.</td>
</tr>
<tr>
<td>FBP_PFETCH_IFETCHQ</td>
<td>INTEGER</td>
<td>The number of prefetches on the immediate fetch queue.</td>
</tr>
<tr>
<td>FBP_PFETCH_IGNORES</td>
<td>INTEGER</td>
<td>The number of prefetches ignored.</td>
</tr>
<tr>
<td>FBP_PFETCH_IO_CNT</td>
<td>INTEGER</td>
<td>The number of prefetch 1/Os.</td>
</tr>
<tr>
<td>FBP_PFETCH_PFETCHQ</td>
<td>INTEGER</td>
<td>The number of prefetches on the prefetch queue.</td>
</tr>
<tr>
<td>FBP_PFETCH_REQ</td>
<td>INTEGER</td>
<td>The number of prefetch requests.</td>
</tr>
<tr>
<td>FBP_POOL_COMPRESS</td>
<td>INTEGER</td>
<td>The number of times pool compress was successful.</td>
</tr>
<tr>
<td>LOGL_AWES_ON_WRITE</td>
<td>INTEGER</td>
<td>The number of AWEs submitted on write.</td>
</tr>
<tr>
<td>LOGL_CHKW_REQUESTS</td>
<td>FLOAT</td>
<td>The number of CHKW requests.</td>
</tr>
<tr>
<td>LOGL_CURR_SEQ_NO</td>
<td>INTEGER</td>
<td>The current log sequence number.</td>
</tr>
<tr>
<td>LOGL_WTBF_CHKPT</td>
<td>INTEGER</td>
<td>The number of wait for buffers during a checkpoint.</td>
</tr>
<tr>
<td>LOGL_WTBF_NOTCHKPT</td>
<td>INTEGER</td>
<td>The number of wait for buffers not during a checkpoint.</td>
</tr>
<tr>
<td>LOGL_WTWT_REQUESTS</td>
<td>INTEGER</td>
<td>The number of WTWT requests.</td>
</tr>
<tr>
<td>PHYL_INTERNAL_CHKW</td>
<td>INTEGER</td>
<td>The number of internal CHKW requests.</td>
</tr>
<tr>
<td>PHYL_OLDS_READS</td>
<td>INTEGER</td>
<td>The number of OLDS reads initiated.</td>
</tr>
<tr>
<td>PHYL_OLDS_WRITES</td>
<td>INTEGER</td>
<td>The number of OLDS writes initiated.</td>
</tr>
<tr>
<td>PHYL_WADS_EXCPVR</td>
<td>INTEGER</td>
<td>The number of WADS EXCPVRs.</td>
</tr>
<tr>
<td>PHYL_WADS_2K_SEG</td>
<td>INTEGER</td>
<td>The number of 2K segment writes to WADS.</td>
</tr>
<tr>
<td>PHYL_WTWT_TIME</td>
<td>FLOAT</td>
<td>The accumulated wait time (all WTWT), in timer units.</td>
</tr>
</tbody>
</table>
## Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI_BYTES</td>
<td>INTEGER</td>
<td>The number of program isolation bytes.</td>
</tr>
<tr>
<td>PI_BYTES_MAX</td>
<td>INTEGER</td>
<td>The maximum number of program isolation bytes.</td>
</tr>
<tr>
<td>PI_SEARCH_CALLS</td>
<td>INTEGER</td>
<td>The number of program isolation search calls.</td>
</tr>
<tr>
<td>PI_SYNONYMS_MAX</td>
<td>INTEGER</td>
<td>The maximum number of program isolation synonym searches.</td>
</tr>
<tr>
<td>PI_SYNONYMS_SEARCH</td>
<td>INTEGER</td>
<td>The number of program isolation synonym searches.</td>
</tr>
<tr>
<td>QP_BUFFER_REPOSN</td>
<td>INTEGER</td>
<td>The number of calls to reposition at the last buffer.</td>
</tr>
<tr>
<td>QP_BUFFERS_LOCKED</td>
<td>INTEGER</td>
<td>The total number of queue pool buffers locked.</td>
</tr>
<tr>
<td>QP_BUFFERS_UNLOCK</td>
<td>INTEGER</td>
<td>The total number of queue pool buffers unlocked.</td>
</tr>
<tr>
<td>QP_BUFFER_WAITS</td>
<td>INTEGER</td>
<td>The number of queue manager wait requests.</td>
</tr>
<tr>
<td>QP_DECB_READ_WAITS</td>
<td>INTEGER</td>
<td>The number of waits for other DECB to read a buffer.</td>
</tr>
<tr>
<td>QP_DECB_WRITEWAITS</td>
<td>INTEGER</td>
<td>The number of waits for other DECB to write a buffer.</td>
</tr>
<tr>
<td>QP_DRRN_HIGH_LMSG</td>
<td>CHAR(8)</td>
<td>The DRRN of the highest long message queue.</td>
</tr>
<tr>
<td>QP_DRRN_HIGH_QBLK</td>
<td>CHAR(8)</td>
<td>The DRRN of the highest queue block.</td>
</tr>
<tr>
<td>QP_DRRN_HIGH_SMSG</td>
<td>CHAR(8)</td>
<td>The DRRN of the highest short message queue.</td>
</tr>
<tr>
<td>QP_ENQDEQ_BFRWAITS</td>
<td>INTEGER</td>
<td>The number of waits for conflicting enqueue-dequeue buffer requests.</td>
</tr>
<tr>
<td>QP_ILOG_WAITS</td>
<td>INTEGER</td>
<td>The number of waits for ILOG.</td>
</tr>
<tr>
<td>QP_IO_ERROR_NORET</td>
<td>INTEGER</td>
<td>The count of temporary I/O errors not retried.</td>
</tr>
<tr>
<td>QP_MSG CANCELS</td>
<td>INTEGER</td>
<td>The number of calls to cancel input or output.</td>
</tr>
<tr>
<td>QP_MSG DEQUEUES</td>
<td>INTEGER</td>
<td>The number of calls to dequeue messages.</td>
</tr>
<tr>
<td>QP_MSG ENQUEUES</td>
<td>INTEGER</td>
<td>The number of calls to enqueue messages.</td>
</tr>
<tr>
<td>QP_PCB_UNCHAINS</td>
<td>INTEGER</td>
<td>The total number of PCBs unchained from buffers.</td>
</tr>
<tr>
<td>QP_PURGE REQUESTS</td>
<td>INTEGER</td>
<td>The number of requests to purge.</td>
</tr>
<tr>
<td>QP_PURGE WAITS</td>
<td>INTEGER</td>
<td>The number of waits for purge completion.</td>
</tr>
<tr>
<td>QP_PURGE WRITES</td>
<td>INTEGER</td>
<td>The number of writes done for purge.</td>
</tr>
<tr>
<td>QP_QMGR_CALLS</td>
<td>INTEGER</td>
<td>The total number of calls to QMGR.</td>
</tr>
<tr>
<td>QP_QMGR LOC ALTERS</td>
<td>INTEGER</td>
<td>The number of locate and alter calls from QMGR.</td>
</tr>
<tr>
<td>QP_QMGR LOCATES</td>
<td>INTEGER</td>
<td>The number of locate calls from QMGR.</td>
</tr>
<tr>
<td>QP_QMGR RELEASES</td>
<td>INTEGER</td>
<td>The number of record release calls from QMGR.</td>
</tr>
<tr>
<td>QP_READ REQUESTS</td>
<td>INTEGER</td>
<td>The number of read requests.</td>
</tr>
<tr>
<td>QP_TRANSLATE_REQ</td>
<td>INTEGER</td>
<td>The number of translate requests.</td>
</tr>
<tr>
<td>QP_WAIT REQUESTS</td>
<td>INTEGER</td>
<td>The number of waits for buffer.</td>
</tr>
<tr>
<td>QP_WRITE REQUESTS</td>
<td>INTEGER</td>
<td>The total number of write requests.</td>
</tr>
<tr>
<td>RECANY_MAX</td>
<td>INTEGER</td>
<td>The maximum number of RECANY buffers used.</td>
</tr>
<tr>
<td>RECANY_USED</td>
<td>INTEGER</td>
<td>The number of RECANY buffers used.</td>
</tr>
<tr>
<td>SCH_BMP_ACTIVE</td>
<td>INTEGER</td>
<td>The number of BMPs active.</td>
</tr>
<tr>
<td>SCH_MPP_ACTIVE</td>
<td>INTEGER</td>
<td>The number of MPPs active.</td>
</tr>
<tr>
<td>SCH_SMB INT CONFL</td>
<td>INTEGER</td>
<td>The counts of SMBs not scheduled because of intent conflict.</td>
</tr>
<tr>
<td>SCH_SMB LOOKED AT</td>
<td>INTEGER</td>
<td>The counts of SMBs looked at for schedule.</td>
</tr>
</tbody>
</table>
## Data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCH_SMB_OTHER</td>
<td>INTEGER</td>
<td>The counts of SMBs not scheduled other than for intent conflict, program conflict, and priority cutoff.</td>
</tr>
<tr>
<td>SCH_SMB_PGM_CONFL</td>
<td>INTEGER</td>
<td>The counts of SMBs not scheduled because of program conflict.</td>
</tr>
<tr>
<td>SCH_SMB_PRIOCUTOFF</td>
<td>INTEGER</td>
<td>The counts of SMBs not scheduled because of priority cutoff.</td>
</tr>
</tbody>
</table>
IMS_CHKPT_VSAM_T

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

The default retention period for this table is 7 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>TIMESTAMP</td>
<td>The full microsecond precision date and timestamp of the time the checkpoint occurred.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the systems programmer. This column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID.</td>
</tr>
<tr>
<td>BUFFER_SIZE</td>
<td>CHAR(8)</td>
<td>The size of buffers in this buffer pool.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>CHAR(8)</td>
<td>The VTAM APPLID for the IMS system. This is the ID by which VTAM communicates with the IMS system. It is derived from the run time parameter :IMS_APPLID (because IMS log records do not contain this field) and selected as the first IMS applid of a summarization group.</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>CHAR(8)</td>
<td>The MVS and JES name of the IMS control region address space. This is derived from the run time parameter :IMS_CTRL_REGION (because few IMS log records contain this field) and selected as the first IMS control region name of a summarization group.</td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>CHAR(4)</td>
<td>The version and release of IMS in which the activity occurred. This is selected as the first IMS version identifier of a summarization group.</td>
</tr>
<tr>
<td>BACKGRND_WRITE_REQ</td>
<td>INTEGER</td>
<td>The number of background write requests.</td>
</tr>
<tr>
<td>BUFFER_ERRORS</td>
<td>INTEGER</td>
<td>The number of buffers currently write-error marked for the IMS session.</td>
</tr>
<tr>
<td>BUFFER_ERRORS_MAX</td>
<td>INTEGER</td>
<td>The maximum number of buffers write-error marked for the IMS session.</td>
</tr>
<tr>
<td>BUFFERS</td>
<td>INTEGER</td>
<td>The number of buffers in the buffer pool for the IMS session.</td>
</tr>
<tr>
<td>CI_FOUND_IN_POOL</td>
<td>INTEGER</td>
<td>The number of times the control interval was found in the subpool for the IMS session.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HS_BUFFERS</td>
<td>INTEGER</td>
<td>The number of hiperspace buffers for the subpool.</td>
</tr>
<tr>
<td>HS_READS</td>
<td>INTEGER</td>
<td>The number of successful VSAM reads from hiperspace.</td>
</tr>
<tr>
<td>HS_READS_FAILED</td>
<td>INTEGER</td>
<td>The number of failed VSAM reads from hiperspace.</td>
</tr>
<tr>
<td>HS_WRITES</td>
<td>INTEGER</td>
<td>The number of successful VSAM writes to hiperspace.</td>
</tr>
<tr>
<td>HS_WRITES_FAILED</td>
<td>INTEGER</td>
<td>The number of failed VSAM writes to hiperspace.</td>
</tr>
<tr>
<td>LRECL_ALTERED</td>
<td>INTEGER</td>
<td>The number of logical records marked altered for the IMS session.</td>
</tr>
<tr>
<td>PLH_WAITS</td>
<td>INTEGER</td>
<td>The number of placeholder waits for the IMS session.</td>
</tr>
<tr>
<td>POOL_ID</td>
<td>CHAR(4)</td>
<td>The name of the buffer pool for the gathered statistics.</td>
</tr>
<tr>
<td>RETRIEVES_BY_KEY</td>
<td>INTEGER</td>
<td>The number of requests to retrieve by key for the IMS session.</td>
</tr>
<tr>
<td>RETRIEVES_BY_RBA</td>
<td>INTEGER</td>
<td>The number of requests to retrieve by RBA for the IMS session.</td>
</tr>
<tr>
<td>SYNC_CALLS</td>
<td>INTEGER</td>
<td>The number of synchronization calls for the IMS session.</td>
</tr>
<tr>
<td>VSAM_ESDS_INSERTS</td>
<td>INTEGER</td>
<td>The number of logical records inserted to ESDS for the IMS session.</td>
</tr>
<tr>
<td>VSAM_GET_CALLS</td>
<td>INTEGER</td>
<td>The number of VSAM get calls issued for the IMS session.</td>
</tr>
<tr>
<td>VSAM_KSDS_INSERTS</td>
<td>INTEGER</td>
<td>The number of logical records inserted to KSDS for the IMS session.</td>
</tr>
<tr>
<td>VSAM_NUSER_WRITES</td>
<td>INTEGER</td>
<td>The number of VSAM space write requests for the IMS session.</td>
</tr>
<tr>
<td>VSAM_READS</td>
<td>INTEGER</td>
<td>The number of VSAM read I/O operations for the IMS session.</td>
</tr>
<tr>
<td>VSAM_SCHBFR_CALLS</td>
<td>INTEGER</td>
<td>The number of VSAM SCHBFR calls issued for the IMS session.</td>
</tr>
<tr>
<td>VSAM_USER_WRITES</td>
<td>INTEGER</td>
<td>The number of VSAM user write requests for the IMS session.</td>
</tr>
</tbody>
</table>
Lookup tables

This section describes the lookup table specific to the IMS Performance feature.

**IMS_APPLICATION**

This lookup table assigns the IMS application for a given transaction to the APPLICATION_NAME column used in the application subcomponent tables. When this table is empty, the default is to assign OTHER to APPLICATION_NAME.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFPRMnn) by the system programmer. For all IMS Performance feature tables, this column is derived from the run time parameter :MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2ImnL supports the specification of an IMSID parameter and uses it to write an 8-byte field at the head of every composite record it creates.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k</td>
<td>The name of the IMS transaction.</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>k</td>
<td>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.</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td></td>
<td>The name of the business application responsible for the transaction processing activity. The name may contain imbedded blanks.</td>
</tr>
</tbody>
</table>

Example of table contents

<table>
<thead>
<tr>
<th>MVS_SYSTEM_ID</th>
<th>IMS_SYSTEM_ID</th>
<th>TRANSACTION_NAME</th>
<th>PROGRAM_NAME</th>
<th>APPLICATION_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS1</td>
<td>IMSP</td>
<td>SE01%</td>
<td>E01%</td>
<td>GENERAL LEDGER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VA%</td>
<td>A%</td>
<td>PAYROLL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VE%</td>
<td>EP%</td>
<td>PENSIONS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>VP%</td>
<td>ER%</td>
<td>PERSONNEL</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TA01</td>
<td>A0100000</td>
<td>ACCOUNT ENQUIRY</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DM%</td>
<td>M0302%</td>
<td>DIRECT MARKETING</td>
</tr>
<tr>
<td></td>
<td></td>
<td>%</td>
<td>%</td>
<td>OTHER</td>
</tr>
</tbody>
</table>

...
Using the GROUP_ID lookup table

When you use the IMS transaction subcomponent, you can reduce the overall collect time (log processing and DB2 updating) by defining the new GROUP_ID lookup table.

You can use the GROUP_ID lookup table when it is enough to have a group for the summarized data and it is not necessary to use the USER_ID, LOGICAL_TERMINAL, and PHYSICAL_TERMINAL key columns in the IMS_USER_TRAN_H, _D, _W table.

GROUP_ID lookup table

Table 9 shows an example of how the table can be coded.

**Primary keys:** USER_ID, LOGICAL_TERMINAL, PHYSICAL_TERMINAL

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_ID</td>
<td>k CHAR(8)</td>
<td>The user identifier used to gain authorized access to IMS resources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column may contain the logical terminal if security is not being managed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>by the IMS-supported SIGN ON and RACF utilities and products.</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td>k CHAR(8)</td>
<td>The IMS-defined logical name for the terminal that is used to request the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transaction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This column is associated with the physical terminal recognized by the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>network.</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td>k CHAR(8)</td>
<td>The network-recognized name for the terminal that is used to request the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transaction. IMS maintains a relationship between this name (which is known</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to the network) and its own logical terminal name.</td>
</tr>
<tr>
<td>GROUP_ID</td>
<td>CHAR(8)</td>
<td>The name of the business resources (for example, the department name) for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the transaction processing activity.</td>
</tr>
</tbody>
</table>

You can use this table to assign the GROUP_ID name for specific USER_ID, LOGICAL_TERMINAL, and PHYSICAL TERMINAL names in the transaction subcomponent.

Before you use this table you must replace any occurrences of the references to USER_ID, LOGICAL TERMINAL, and PHYSICAL TERMINAL with the GROUP_ID column. This has to be done for all the objects (tables and update definitions) included in the transaction subcomponent in which any of the above-mentioned columns appear. See the Language Guide and Reference and the Administration Guide for more information.
Part 2. IMS Shared Queue feature

Chapter 7. Introduction to the IMS Shared Queue feature

Understanding the IMS Shared Queue feature ........................................ 103
Collecting data ...................................................................................... 104
SLDS ................................................................................................. 104
Log procedure ...................................................................................... 105
Composite record ................................................................................. 105
Record procedure ................................................................................ 105
Record definitions ................................................................................ 105
Tivoli Decision Support for OS/390 data tables and environmental information ........................................ 105
Reports .................................................................................................. 105
The log collector and DRLSLOGP ......................................................... 105
Installing and customizing the IMS Shared Queue feature ............... 106
Planning for the IMS Shared Queue feature ......................................... 106
Selecting IMS Shared Queue feature components .............................. 107
The collect components ....................................................................... 108
Updating other lookup and control tables ........................................... 108
Using the IMS Shared Queue feature .................................................. 108

Chapter 8. Using log and record procedures within the IMS Shared Queue

The log procedure .................................................................................. 109
Set relationships .................................................................................. 109
Composite records and subtypes .......................................................... 110
Handling of special IMS cases within the IMS Shared Queue .......... 110
Log procedure DRLOUT reports ......................................................... 113

Chapter 9. Understanding data flow through IMS Shared Queue

Log collector data flow ......................................................................... 116
DRLSLOGP data flow ......................................................................... 117
IMS Shared Queue Collect feature data flow ....................................... 118

Chapter 10. Administering the IMS Shared Queue feature

IMS Shared Queue Logs ...................................................................... 121
Specifying DRLSLOGP and log procedure parameters ....................... 121
Specifying the log collector parameter ................................................. 124
Running the log collector in a shared queue environment .................. 125
Using DRLSLOGP .............................................................................. 127
DRLSLOGP input and output data sets ............................................... 127
Running DRLSLOGP .......................................................................... 128
Operational considerations .................................................................. 129
Running the IMS Light feature ............................................................. 130
Setting up Load Library for the IMS Light feature ............................ 131
Statement description .......................................................................... 131
Setting up the Tivoli Decision Support collect for the IMS Light feature ........................................ 132

Chapter 11. IMS Shared Queue record definitions

Composite record definitions ................................................................. 133
Composite record sections ................................................................. 133

Chapter 12. IMS Shared Queue data tables and lookup tables

IMS Shared Queue data tables .............................................................. 135
Transaction Transit Time subcomponent tables and views ................. 135
Data tables .......................................................................................... 136
IMSS_TRAN_H_D_W ........................................................................ 136
IMSS_TRAN_QUEUE_Q_D ............................................................. 140
IMSS_SYSTEM_TRAN_H_D ............................................................ 143
TRANS_TYPE key column ................................................................. 145
View .................................................................................................... 147
IMSS_TRAN_QUEUE_QV_DV ......................................................... 147
Account and Availability subcomponent tables .................................. 148
Data Tables ......................................................................................... 148
IMSS_FSB_ACCOUNT_H_D_W ...................................................... 148
Availability for IMS resources ............................................................. 153
IMSS_AVAILABILITY_T ................................................................. 154
IMSS_AVAILABILITY_D_W ............................................................ 155
Lookup tables ...................................................................................... 155
IMSS_AVAILRESOURCE ............................................................... 155
Chapter 7. Introduction to the IMS Shared Queue feature

Tivoli Decision Support for OS/390 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 OS/390 consists of a base product and several optional features.

The Tivoli Decision Support for OS/390 base includes:

- Interactive System Productivity Facility (ISPF) host reporting and administration dialogs
- The Tivoli Decision Support for OS/390 log collector program
- Log and record definitions for all records used by the Tivoli Decision Support for OS/390 features

Each feature provides:

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

Tivoli Decision Support for OS/390 enables you to collect large volumes of data and keep the space to store it at acceptable levels. This Tivoli Decision Support for OS/390 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 OS/390 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.

When you use IMS 6.1 and IMS 7.1 in a non-shared Queue configuration, you can decide whether to use the pre-Shared Queue support engine (not including the latest enhancements/re-design implementation) or the new Tivoli Decision Support for OS/390 SQ engine (even if you are in a non-SQ IMS environment). With the IMS 8.1 support, you can only use the IMS SQ Collect Component. In order to collect data from IMS 8.1, you must use the new engine with the CSQ_V810_COLLECT process. This means that the pre-Shared Queue support IMS tables will not be updated with the IMS 8.1 data anymore, as only the new tables provided by the IMS Shared Queue support will be used. The checkpoint statistical data, the product’s IMS tables remain unchanged.

Understanding the IMS Shared Queue feature

The IMS Shared Queue feature collects IMS performance data to produce reports. Reports are produced using information stored in Tivoli Decision Support for OS/390 DB2 tables. Figure 1 on page 4 shows an overview of the IMS Shared Queue feature.
Collecting data

The process of collecting IMS performance data into DB2 tables is called a *collect*. It works like this:

**SLDS**

Each IMS system that is a member of the same IMS shared group produces a system log data set (SLDS) during the IMS archive process. All these logs are then merged into a single SLDS, that is used as input by the IMS Shared Queue feature, or can be read sequentially by using the online merging option.

---

*Figure 18. Overview of data collection using the IMS Shared Queue feature*
Log procedure
A Tivoli Decision Support for OS/390 log processing program called a log procedure processes selected records in the merged SLDS. The log procedure matches records that have been written for IMS events. An IMS event is an activity that is part of a transaction or an IMS system activity. Each IMS record type represents an IMS event. Together, a number of records contain all information about a transaction.

The log procedure saves the matched records until the transaction is complete.

Composite record
The log procedure creates a composite record from these matched records in the merged SLDS log—when the transaction is complete.

Record procedure
A Tivoli Decision Support for OS/390 record processing program called a record procedure processes the composite records and creates simplified records called R2 records.

Record definitions
The IMS Shared Queue feature provides record definitions for the individual record types found in the merged IMS SLDS, and for the additional records created by the Tivoli Decision Support for OS/390 record procedure. The record definitions are used by Tivoli Decision Support for OS/390 collect when updating DB2 tables.

Tivoli Decision Support for OS/390 data tables and environmental information
The IMS Shared Queue feature uses the R2 records, along with user-supplied data in Tivoli Decision Support for OS/390 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.

Reports
Tivoli Decision Support for OS/390 creates reports from the information in the data tables. In addition to the reports provided with the IMS Performance feature, you can create your own reports using, for example, the Query Management Facility (QMF) prompted query language.

The log collector and DRLSLOGP
To collect data as described in Figure 18 on page 104 you run the Tivoli Decision Support for OS/390 log collector program. The log collector uses record definitions and other definitions when it updates Tivoli Decision Support for OS/390 data tables. The log collector is part of the Tivoli Decision Support for OS/390 base product.

An alternative way to process the merged SLDS without using the log collector is to run a batch program provided with the IMS Shared Queue feature, called DRLSLOGP.

DRLSLOGP is a stand-alone batch program that calls the log procedure and record procedures. DRLSLOGP does not update the DB2 tables. It produces output composite records.

DRLSLOGP is usually used to obtain an input log (DRLICOMP) shorter than the IMS original IMS log. You do not have to install the IMS Shared Queue feature or its components to use DRLSLOGP.
Installing and customizing the IMS Shared Queue feature

This section supplements the general feature installation procedure described in the Administration Guide for installing and customizing a Tivoli Decision Support for OS/390 feature component.

To install and use the IMS Shared Queue feature, you must have an MVS operating system capable of running Tivoli Decision Support for OS/390. The IMS Performance feature supports data from systems running IMS/ESA from Version 6 Release 1 to Version 8 Release 1. You use the pre-merged or online-merged IMS system log data set (SLDS) to generate data for the predefined tables and reports in the IMS Shared Queue feature. You can also use each SLDS system separately, but some data, such as timing information, might become inaccurate.

Figure 19 shows the sequence of events in planning for, installing, customizing, and administering a Tivoli Decision Support for OS/390 feature.

Planning for the IMS Shared Queue feature

Your most critical planning task is determining what kind of information users need from the IMS Performance feature. For example, users may be interested only in system resource availability or transaction response time. Installing only those...
parts of the feature needed to meet user requirements ensures that the feature benefits users while minimizing the performance impact caused by data collection and interpretation activities.

After you have installed the product using SMP/E, plan each step of the implementation process:

1. Determine what users need from the IMS Shared Queue feature. What tasks must they perform that the feature can accomplish or assist with?

2. Determine what components and subcomponents you must install to meet users' needs. See "Selecting IMS Shared Queue feature components" for a description of the components and subcomponents available.

3. Determine the administration tasks you must perform to customize Tivoli Decision Support for OS/390 and the IMS Shared Queue feature to work with your computer system. Make any decisions necessary to perform these tasks.

4. For the selected components, determine the customization tasks required to customize the supported products to work with Tivoli Decision Support for OS/390 and with the IMS Shared Queue feature.

If you are planning for the first time, you must perform all these steps to ensure that your implementation of the feature is consistent and is driven by a common goal. If you are reading this chapter in preparation for modifying your system, you may not need to perform all of these tasks.

The detailed planning tasks you must perform depend on the components you choose to install. However, the basic planning process is the same for all components.

When you are ready to install and customize an IMS Shared Queue feature component, refer to the procedures in the Administration Guide.

Selecting IMS Shared Queue feature components

The IMS Shared Queue feature is divided into components and subcomponents. Consider carefully which of these to install. If you need reports from a component that you have not installed, you must install the component and then wait several days or weeks until enough data has been collected to create reports. Alternatively, if you install more components than you need, Tivoli Decision Support for OS/390 collects unnecessary data, which takes up disk space.

The IMS Shared Queue feature components and subcomponents contain Tivoli Decision Support for OS/390 objects (for example, predefined reports, tables, and update definitions). Each IMS Shared Queue feature component contains the objects required to collect performance and service level data from the appropriate records in the merged IMS log and produce reports.

You can install all Tivoli Decision Support for OS/390 features and components using the procedure in Administration Guide. After the system programmer has successfully installed the Tivoli Decision Support for OS/390 base, you can choose whether to install the IMS feature and its components and subcomponents. Tivoli Decision Support for OS/390 stores the necessary log, record, and update definitions in Tivoli Decision Support for OS/390 system tables. Tivoli Decision Support for OS/390 also loads predefined DB2 tables and reports.

The IMS Shared Queue feature components are:

- IMS 6.1 CSQ Collect Component, for IMS Version 6
Introduction to the IMS Shared Queue Feature

- IMS 7.1 CSQ Collect Component, for IMS Version 7
- IMS 8.1 CSQ Collect Component, for IMS Version 8

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

Account and Availability subcomponent
Collects information about resource consumption and availability data about the IMS systems, regions, and applications.

Statistics subcomponent
Records statistical information about buffer and pool usage.

Updating other lookup and control tables
The IMS Shared Queue feature uses the DAY_OF_WEEK and PERIOD_PLAN control tables, which are also used by other Tivoli Decision Support for OS/390 features. Check these tables and update them as needed.

For information about these tables, refer to the Administration Guide.

If you have installed the Tivoli Decision Support for OS/390 System Shared Queue feature, you can use it to collect and report on data regarding 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 about the table, refer to the System Performance Feature Reference Volume I.

Using the IMS Shared Queue feature
Before starting the daily use of the IMS Shared Queue feature, run a few tests to ensure that the installation was successful. Verify that Tivoli Decision Support for OS/390 is collecting the right data, storing the data correctly, and using the proper data to generate the reports. Verify also that the lookup table contains the appropriate groups.

After you verify that the installation was successful, you can put the IMS Shared Queue feature into production.

Refer to the Administration Guide for the steps in testing component installation and for general instructions for running Tivoli Decision Support for OS/390. For specific information about running the IMS Shared Queue feature, see Chapter 10, "Administering the IMS Shared Queue feature", on page 121.
Chapter 8. Using log and record procedures within the IMS Shared Queue

This chapter explains the use of log procedures and record procedures within the IMS Shared Queue feature. A log procedure takes two or more records from a log and creates one 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 DRLSI612, the DRLSI712, and DRLSI812 record procedures take composite records created by the log procedure and simplify them to make collection and reporting easier. You can also add record procedures that can be used for different purposes.

The log procedure

The IMS Shared Queue feature is based on a log processing routine (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), rather than at the program specification block (PSB) level, and therefore the records are more detailed and meaningful. The log procedure copies the most relevant IMS log record fields to the composite record. To keep track of the complete IMS records sequence, the IMS Common Shared Queue log collector is based on an IMS key named unit of work (UOW), that is unique through all the IMS logs and in the IMS merged log. The unit of work key is valid in shared and not shared IMS configurations. The recovery token is still used to link the PSB.

Set relationships

Depending on the type of transaction, certain relationships exist between the different logical sets of record groups. From IMS Version 6, a unit of work (UOW) key is used to uniquely identify a message starting from the X’01’ log record and may be used to tie together related X’03’ log records. The UOW is a field of 32 bytes with the following format:

Originating-system message ID

The message ID assigned by the IMS (OUOW: first 16 bytes).

Processing-system message ID

The message ID assigned by the IMS system that processed the message (PUOW: last 16 bytes).

Because the UOW comprises IDs for both the system that originates the message and the system, if any, that processes the message, all messages that are associated with an original message can be tied together by the original unit of work (OUOW) key. The processing unit of work (PUOW), instead, can change during the different steps of the transaction life.

(OUOW-PUOW)

The link between OUOW and PUOW is valid for all kinds of transaction. This link is determined indirectly through the presence of the same 16-byte Origin Unit Of Work on a X’01’ record for input PUOW or X’03’ record for
Using lof and record procedures within the IMS Shared Queue

output PUOW. The IMS records X'35', X'31', X'36', and so on will be
chained to input PUOW or output PUOW depending on the processing
unit of work key value.

(Input PUOW-PSB)
The X'31' In of the input PUOW set serves to create the link, as it carries a
16-byte recovery token whose high-order 12 bytes are also the PSB set key.

(Output PUOW-PSB)
The X'35' (sometimes also the X'31' APPL SYNC) Out of the output PUOW
set serves to create the link, as it carries the 16-byte recovery token whose
high-order 12 bytes are also the PSB set key.

Composite records and subtypes
The IMS Shared Queue log collector produces only one composite record type
X'FF' that will be used as input by the IMS record procedure DRLSnn2 (where nn
is 61, 71, or 81). Another type of composite record is created using X'06', X'07',
X'08', X'0A07' and X'45', X'47', X'4001' checkpoint records that will be used directly
by the Tivoli Decision Support for OS/390 log collector to populate the tables
provided without invoking any IMS record procedure. The log procedure can write
composite records for incomplete transactions when limits specified by the
TABLEFLUSH or WRITEPENDING parameters have been reached.

Handling of special IMS cases within the IMS Shared Queue
Within the IMS Shared Queue feature, the log procedure handles special IMS cases
as described hereafter.

Multiple segment input
The first or only segment creates an Input-OUOW/PUOW table entry. The
log procedure skips subsequent segments.

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

Multiple segment output
The first or only segment X'03' creates an Output-PUOW table entry.

Multiple outputs
Each output creates an Output-PUOW table entry. Multiple outputs are
valid only when a corresponding input set or PSB set is present. The
multiple outputs are linked through pointers, and each new output is
placed at the end of the current list. When the log procedure writes a
composite record, it also searches the linked list and writes all output sets
for a given input set. Multiple linked outputs are possible only when
outputs are from a scheduled PSB (PSB set is present). Because the
procedure attempts to produce composite records at transaction level
rather than at PSB level, it links the outputs to an input message set
OUOW.

Multiple transactions per schedule of a PSB
The log procedure creates the RTKN entry upon receiving a X'08' record.
The X'31' In record, which carries the PUOW and the recovery token of the
PSB set, creates the linkage between an input-PUOW table entry and the
corresponding RTKN entry. The composite record for a completed
transaction is written as soon as the X'33' record is found, but the PSB control block will be kept and freed only when the X'07' record is encountered.

Program-to-program switch

Distinctions are made between transactions that started with a X'01' record (root transaction) and those that started with a program switch (child transaction), as indicated by a X'03' record that has the MSGQDES flag set to X'81' (destination is an SMB) rather than X'82' (destination is a CNT). Although the log procedure will write the root transaction composite record when it is considered complete, it retains all the input set details until all child transactions created by the root have also been completed.

For example, if A is a root transaction that creates transaction B, and B creates transaction C, C is also treated as a child transaction created from A. The log procedure retains the root transaction’s input set details, because it writes the input set of the root transaction for the child transaction as well as its own input set. Differently from the old engine, the root transaction’s input set details will be subject to the TABLEFLUSH parameter. The OUOW control block contains a list of pending child transactions. For the child transaction, the input-PUOW control block is pointed by to root transaction PUOW.

AOI user-exit initiated transactions

The log procedure treats the X'03' record corresponding to the AOI user exit like a X'01' record and builds an input-OUOW/PUOW control block.

Output message reenqueue

Here, (for example, when IMS finds that the terminal does not acknowledge successful receipt of a message) IMS can:
1. Save the output message (indicated by a X'36' save record)
2. Reenqueue the same message to the same destination (indicated by the appearance of a second X'35' record with the reenqueue flag set)
3. Get the unique message from the output queue again (indicated by the appearance of a second X'31' record)
4. Dequeue the message, if the terminal acknowledges successful receipt of the message (indicated by the presence of a second X'36' record)
5. Delete the message from the queue (indicated by the appearance of a X'33' 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 GU; thus the delay would be attributed to the network.

Message-driven BMP programs

These are treated exactly like full function transactions.

Non-message-driven BMP programs

Will not be handled by the record procedure but will only be used to populate the IMS_PSB_ACCOUNT_x tables.

System-generated output (including master terminal operator (MTO) traffic)

The X'03' record creates output-OUOW/PUOW entries. When the log procedure receives the X'36' (DEQ) or X'33' (GET) records, it writes the output-OUOW/PUOW entry as a composite record.

Terminal message switch

The X'01' creates the input-OUOW/PUOW entries. When the log procedure
receives the X’33’ (GET) record for the input PUOW, it writes the
tinput-PUOW entry as a composite record. This special case may also
include MSC and ISC message switching.

Conversational transactions
The IMS records X’11’, X’12’, and X’13’ are no longer supported.

Conversational transactions with program-to-program switch
The IMS records X’11’, X’12’, and X’13’ are no longer supported.

Fast Path (EMHs)
The X’5901’ record creates EMH OUOW/PUOW entries. The log procedure
matches subsequent X’5903’ records and X’5936’ records to the EMH entry
using the UOW and recovery token keys, if present.

MSGQ output produced by EMH transactions (if any) links through the
16-byte recovery token carried by the X’35’ Out record of output PUOW.

Wait-for-input (WFI) programs
If a region is processing pseudo WFI or WFI transactions with a high
PROCLIM value you may not get the X’07’ records that are required to
produce certain statistics. For this reason you may want to consider
reducing the PROCLIM value (it can still be large) or scheduling a new
region so that the X’07’ records get logged. If the regions and transactions
are properly classed then a ‘Quick Reschedule’ will occur such that the
X’07’ and X’08’ records are created but the program is never truly
terminated and rescheduled. Alternatively you need to include all logs
until the relevant regions terminate. In any case, if the program is a WFI or
PWFI, the log procedure does not hold an incomplete transaction until the
X’07’ record is encountered. Instead, the log procedure writes the composite
record when all PUOW entries for the corresponding OUOW are
completed (received X’33’ records). The log procedure approximates the
process time as the difference between X’35’ Out Time and X’31’ In Time.
The log procedure does not delete the PSB control block until the X’07’
record is encountered. If the log procedure cannot determine that the
program is a WFI, it assumes multiple transactions for a single PSB
schedule. Differently from the old IMS Log procedure engine, the X’07’
record is handled independently and it is able to populate directly the
IMS_PSB_ACCOUNT_\text{x} tables providing metrics that otherwise the user
would lose. TABLEFLUSH parameter can cause pending transactions to be
flushed.

Quick reschedule
Quick reschedule is not an option. It 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.
Quick reschedule allows application programs to process more than the
processing limit of messages for each physical schedule. It eliminates the
processing overhead caused by unnecessary rescheduling and reloading of
application programs.

PROCLIM 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 pseudo-WFI. If PROCLIM=65565, the number of messages that can be processed per scheduling is unlimited. The X'07' record indicates a case of quick reschedule. Because IMS may not write the subsequent X'08' record, the log procedure creates the PSB entry upon receiving the X'07' record. The X'08' record, if it follows, is matched to the already-created PSB entry.

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

The IMS records written here 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 16.

CPI-C Saa Driven Application Programs

The CPI-C program termination record X'0A07' is now handled by the IMS Shared Queue log collector and used to populate the tables IMS_PSB_ACCOUNT_x reporting resources consumption. The rows in IMS_PSB_ACCOUNT_x tables resulting from CPI_C Saa driven application programs are recognizable for having the column PROGRAM_TYPE='CPI REGION'. 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 cause transactions in a MPP Region are traced in the Tivoli Decision Support for OS/390 IMS_TRAN_H table and they are recognizable for having the key column TRANS_TYPE = 'M---S-C-' (Please refer to TRANS_TYPE explanation in the data table “IMS_SYSTEM_TRAN_H,_D” on page 143).

Synchronous APPC(OTMA) Conversations

There are no X'35' Out 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 records flow is X03-X31-X33. The X'31' record is a special ipbypass enqueue GU record, as designated by the QLGU1NOE bit. Synchronous conversations support has been added with IMS Version 8 Release 1 support.

Log procedure DRLOUT reports

During normal processing, the log procedure produces several useful reports, and informational, warning, and error messages. For information about messages and codes issued by the log procedure, refer to the Messages and Problem Determination manual.

The log procedure parameter report [Figure 20 on page 114] 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.
The log procedure pending node report (Figure 21) appears after the log procedure has completed and indicates the number of nodes pending in storage tables by type. Nodes are the internal representations of the data before they are grouped as a complete composite record. The log procedure writes these pending nodes to the checkpoint file allocated to DRLICHKO, (if present) which can be used when processing the next SLDS for the same IMS system or the next merged SLDS from different IMS systems in shared queue configuration.

Figure 20. Example of log procedure parameter report within IMS Shared Queue

The log procedure pending node report (Figure 21) appears after the log procedure has completed and indicates the number of nodes pending in storage tables by type. Nodes are the internal representations of the data before they are grouped as a complete composite record. The log procedure writes these pending nodes to the checkpoint file allocated to DRLICHKO, (if present) which can be used when processing the next SLDS for the same IMS system or the next merged SLDS from different IMS systems in shared queue configuration.

Figure 21. Example of log procedure pending node report within IMS Shared Queue

The log procedure composite record report no longer appears, because the IMS Shared Queue feature has only one composite record.
This chapter describes the flow of data through the IMS Shared Queue Collect feature. The chapter includes:

- Tivoli Decision Support for OS/390 data flow overview
- IMS Shared Queue Collect feature data flow through the log and record procedures
  - When collecting data into DB2 tables using the Tivoli Decision Support for OS/390 log collector
  - When generating output composite records using the DRLSLOGP batch program
- IMS Shared Queue Collect feature data flow through the collect component
Log collector data flow

Figure 22 shows the flow of data from the SLDS to the DB2 tables when you use the IMS Shared Queue log collector. For more information about running the log collector, see “Running the log collector in a shared queue environment” on page 125.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>The IMS log contains the original data as written by IMS, or the log that was created by merging the SLDSs written by IMS systems that are members of the same IMS shared group. The ddname DRLLOG points to the data set you want to process.</td>
</tr>
<tr>
<td>2.</td>
<td>The log collector calls the log procedure (DRLSinnL, where nn is the level of IMS you are using; for example, DRLSI61L for IMS Version 6 Release 1).</td>
</tr>
<tr>
<td>3.</td>
<td>The log procedure calls a parameter-checking procedure to verify the processing options you have 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.</td>
</tr>
<tr>
<td>4.</td>
<td>At key commit times while the log collector is running, the log procedure writes to the checkpoint file DRLICHKO. You can use DRLICHKO 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. DRLICHKO becomes DRLICHKI the next time you use the log procedure.</td>
</tr>
<tr>
<td>5.</td>
<td>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.</td>
</tr>
<tr>
<td>6.</td>
<td>The log collector sends the composite records to the record procedure, where the record procedure simplifies the records for easier collection and reporting.</td>
</tr>
</tbody>
</table>

Figure 22. IMS Shared Queue data flow through the log collector
7. The log collector uses the extract records from the record procedure, the composite records from the log procedure.

**DRLSLOGP data flow**

Figure 11 on page 27 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 127.

The characters \( nn \) indicate the IMS release and can be 61, 71 or 81.

---

**Figure 23. Data flow through DRLSLOGP**

DRLSLOGP goes performs steps to create output:

1. The IMS log contains the original data as written by IMS, or the log that was created by merging the SLDSs written by the IMS systems that are members of the same IMS shared group. The ddname is DRLLOG, which points to the data set you want to process.

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 (DRLSInnL, where \( nn \) is the level of IMS you are using; for example, DRLSI71L for IMS Version 7 Release 1).

4. The log procedure calls a parameter-checking procedure to verify the processing options you have 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. After the log procedure finishes processing, it writes the remaining unmatched transaction records to DRLICHKO. DRLICHKO becomes 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. The log procedure does all of the matching and processing to create the composite records, and sends the composite records back to DRLSLOGP.

7. The DRLSLOGP sends the selected records to a record procedure, which can simplify the records for easier collection and reporting. You can specify only one record procedure, the IMS Shared Queue Collect feature writes the output to data set DRLIRPT2. There could be n outputs from a record procedure for the same composite record. You can specify the output data set as disk, tape, or dummy output.

8. DRLSLOGP sends the composite records to data set DRLICOMP, which you can specify as disk, tape, or dummy output.

**IMS Shared Queue Collect feature data flow**

The characters n stand for the IMS release and can be 61, 71 or 81 and the characters nnn stand for the IMS release and can be 610, 710 or 810.

---

**Figure 24. Data flow—Account and Availability subcomponent**
Figure 25. Data flow—Transaction Transit Time subcomponent
Chapter 10. Administering the IMS Shared Queue feature

This chapter explains how to use the IMS Shared Queue feature to process and collect IMS data. You can use the log collector program alone or you can use DRLSLOGP in batch mode followed by the log collector, if you want to collect the resulting data. For more information about Tivoli Decision Support for OS/390 administration, refer to the Administration Guide.

IMS Shared Queue Logs

In a SYSPLEX IMS environment each IMS System has its own SLDS Log where the transaction records are logged.

Tivoli Decision Support for OS/390 reads the Log records, recreates the transaction flow, and evaluates the IMS statistics, saving the data into the DB2 Tables of the IMS feature. When working with Shared queue, each IMS continues to write its own log so that the records for a "unique" transaction can be spread across several logs. Tivoli Decision Support for OS/390 needs to have the whole cumulative log in order to recreate the complete transaction across the IMS SQ systems.

The options provided by Tivoli Decision Support for OS/390 with IMS Shared Queue Support are:

1. To merge all the logs into only one log using the merge utility provided by IMS (DFSLTMG0). This means that the IMS logs are pre-processed before the Tivoli Decision Support for OS/390 collect is run. The maximum number of logs merged by the IMS utility is nine. See Appendix E, “DFSLTMG0 log merge utility”, on page 231 for details.

2. To use the online merging option of Tivoli Decision Support for OS/390 using the SQNLOGS parameter. This means that in input Tivoli Decision Support for OS/390 will open all the logs from the different IMS systems defined in the collect JCL. It will read, in turn, the first record of each log and decide which one is the next record to be processed. This avoids the pre-processing of logs, but adds an overhead to the overall collect process. The maximum number of logs that can be merged online is nine. See “Running the log collector in a shared queue environment” on page 125 for details. The collect terminates with the log that ends first. In both cases, the logs processed must be of the same IMS version and refer to the same time interval.

Specifying DRLSLOGP and log procedure parameters

You can specify parameters to control the operation of DRLSLOGP and the log procedure. Table 3 on page 31 and Table 4 on page 31 list these parameters, with a brief description, the default, and a guideline as to the 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 Version 6 Release 1 is specified as

IMSVER=61
## Administering the IMS Shared Queue feature

### Table 10. Parameter summary for DRLSLOGP

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Possible values</th>
<th>Default</th>
<th>Description</th>
<th>Performance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSVER</td>
<td>61, 71, 81</td>
<td>61</td>
<td>IMS version and release number.</td>
<td>None</td>
</tr>
<tr>
<td>REPORTS</td>
<td>R2</td>
<td>None (produce no reports)</td>
<td>Report programs (or record procedures) that should be run.</td>
<td>None</td>
</tr>
</tbody>
</table>

### Table 11. Parameter summary for the log procedure within IMS Shared Queue

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>Possible values</th>
<th>Default</th>
<th>Description</th>
<th>Performance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSID</td>
<td>Valid subsystem IDs, up to 8 characters long</td>
<td>IMS</td>
<td>The IMS subsystem ID.</td>
<td>None</td>
</tr>
<tr>
<td>MAXFREE</td>
<td>Whole numbers between 800 and max. system capacity</td>
<td>800</td>
<td>Sets the number of internal buffers for queueing incomplete transactions.</td>
<td>High</td>
</tr>
<tr>
<td>RECTYPE</td>
<td>Valid hex numbers from 00 to FF</td>
<td>FF</td>
<td>Record type of composite records created.</td>
<td>None</td>
</tr>
<tr>
<td>RESTARTCHECK</td>
<td>NO</td>
<td>None</td>
<td>Specifies whether the collect restarts if the IMS restart record is found. The collect restarts form the first valid record after the IMS restart record.</td>
<td>Low</td>
</tr>
<tr>
<td>SQNLOGS</td>
<td>Integer numbers between 1 and 9</td>
<td>1</td>
<td>Controls the number of logs that will be opened in input Medium - High (according to the number of logs and record sequences)</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>Valid values for yyyy-mm-dd-hh:mm:ss.t</td>
<td>Start of log</td>
<td>The IMS log date and time from which to start processing.</td>
<td>None</td>
</tr>
<tr>
<td>STOP</td>
<td>Valid values for yyyy-mm-dd-hh:mm:ss.t</td>
<td>End of log</td>
<td>The IMS log date and time at which to stop processing.</td>
<td>None</td>
</tr>
<tr>
<td>TABLEFLUSH</td>
<td>Whole numbers between 1 and 9999</td>
<td>None</td>
<td>Age limit (in seconds) of table entries. Entries older than this limit are flushed from the table when an IMS checkpoint is encountered or the storage limit is reached.</td>
<td>Medium</td>
</tr>
<tr>
<td>WRITEPENDING</td>
<td>NO</td>
<td>NO</td>
<td>Controls whether pending table entries (those not completed at the end of the log procedure) are written as output. If the DRLICHKO ddbname is present, this parameter is ignored.</td>
<td>Low</td>
</tr>
<tr>
<td>ACCOUNT</td>
<td>YES NO</td>
<td>YES</td>
<td>Controls whether X'06', X'07', X'08' and X'0A07' records will be written in DRLICOMP dataset as CSQ_Vnn_STxx records.</td>
<td>Low</td>
</tr>
<tr>
<td>STATISTIC</td>
<td>YES NO</td>
<td>YES</td>
<td>Controls whether X'40cx7' records will be written in DRLICOMP dataset as CSQ_Vnn_STxx records.</td>
<td>Low</td>
</tr>
</tbody>
</table>

The parameters for DRLSLOGP are:
IMSVER=nn
Specifies which release of IMS log data you are using as input, where nn is
the version and release number. For example, if you are using IMS Version
8 Release 1 data, specify IMSVER=81. The default is 61, which specifies IMS

REPORTS=R2(xx-yy)
Specifies which report programs (or record procedures) to run when the
IMS Shared Queue feature encounters a particular record type. R2
identifies the record procedure and xx-yy identifies the record type. For
example, to run DRLSI nn2 when the IMS Shared Queue feature encounters
record type XFF, specify REPORT=R2(FF). The default is to produce no
reports.

The parameters for the log procedure within the IMS Shared Queue are:

IMSID=xxxxxxxx
Optional. Specifies the IMS subsystem name. The IMS Shared Queue
feature is able to get it from the records. xxxxxxxx is the name of the IMS
subsystem. For example, if your IMS subsystem name is IMS2, specify
IMSID=IMS2.

MAXFREE=nnnn
Specifies the number of internal buffers that will be used to queue
incomplete transactions. Do not set this parameter to a value which is less
than 800. Values under 800 will cause queueing problems because of
inadequate number of internal buffers. If queueing problems occur,
increase the value in 50% increments until the problem is resolved.
However a too large increase in the parameter value may result in
excessive system memory being assigned to the internal buffers. Possible
values are from 800 to max. system memory space. The default is 800.

RECTYPE=nn
Specifies the record type for the composite records, where nn is the record
type. For example, if you want the composite records to have type X’FA’,
specify RECTYPE=FA. The default is FF.

RESTARTCHECK=NO
If you specify No, when an IMS restart record is found, Tivoli Decision
Support for OS/390 does not stop execution. A DB2 commit is performed
for all transactions that are consistent and the IMS collects and restarts by
using the valid records after the IMS restart record.

SQNLOGS = n
Specifies the number of IMS Logs that will be opened for input. This
number should match the number of DRLLOG(x) DD statements specified
in the collect or DRLSLOGP job. Valid values are between 1 and 9. The
default is 1, it will preserve the previous way of handling the IMS log
read.

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.

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 the number of seconds that an entry can remain in log procedure internal tables. The log procedure removes entries older than this limit from the table when an IMS checkpoint is encountered or the storage limit is reached. If you do not specify a TABLEFLUSH value, table entries accumulate indefinitely and the tables are not flushed. For example, if you specify TABLEFLUSH=5 and the storage limit is reached, transaction entries that started more than 5 seconds before the date and time found in the last processed relevant record are removed from the table.

WRITEPENDING=xxx
Specifies whether the log procedure writes unmatched table entries to an output record when it is finished processing. If the DRLICHKO ddname is present, the log procedure ignores this parameter. Possible values are NO and YES. The default is NO.

ACCOUNT=xxx
Specifies X'06', X'07', X'08', and X'0A07' records will be written in DRLICOMP dataset as CSQ_Vnn_STxx records. If you do not install the Account and Availability subcomponent, set ACCOUNT=NO. The default is YES.

STATISTIC=xxx
Specifies X'45' records will be written in DRLICOMP dataset 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.

Specifying the log collector parameter

When running the log collector, you specify log collector parameters and COLLECT statements using in-stream JCL or in a data set allocated to ddname DRLIN. The format for specifying these parameters is: the keyword SET in column 1, followed by the parameter name, an equal sign (=), the value in single quotes ('), and ending with a semicolon (;). For example, the IMS_SYSTEM_ID of IMS1 is specified as:

SET IMS_SYSTEM_ID = 'IMS1';

Refer to the Language Guide and Reference for a description of the COLLECT and SET statements. Specify the following parameters for IMS collect:

SYSPLEX_NAME
Specify the Sysplex name from the SYSPLEX parameter in theCOUPLExx parmrib member). The IMS Shared Queue feature uses this value to populate a non key column in tables. If you do not specify this parameter, the column contains the value $UNKNOWN. Set this field in collecting IMS logs from shared queue environment.

MVS_SYSTEM_ID
The ID for the MVS system where IMS was running. The IMS Shared Queue feature uses this value to populate a non key column in the tables, so specify it with care. If you do not specify this parameter, the column contains the value $UNKNOWN.

IMS_SYSTEM_ID
The ID for the IMS system. The IMS Shared Queue feature uses this value
as an alternative to that extracted by the IMS records. It is used in a key column. Do not set externally IMS_SYSTEM_ID collecting IMS logs from shared queue environment.

Running the log collector in a shared queue environment

You can run collect online using the administration dialog, or in batch. To run collect, refer to the Administration Guide.

Figure 26 shows an example of how to run an IMS Shared Queue collect.

```bash
//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 OS/390 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 SYSCOUT=*,DCB=(LRECL=80)
// DRLDUMP DD SYSCOUT=*,DCB=(LRECL=32756)
// DRLIPARM DD *
ACCOUNT=YES
MAXFREE=3000
//DRLIN DD DISP=SHR,DSN=USERIDA.IMS.DEFS.V61(DRL$CVAR)
// DD *
COLLECT CSQ_V610_COLLECT -- IMS V6 or CSQ_V710_COLLECT for IMS V7
BUFFER SIZE 50 M; -- Appropriate collect buffer size
```

Figure 26. Sample job for running the log collector within IMS Shared Queue

To run the log collector for IMS V8.1, you have to change the collect statement into CSQ_V810_COLLECT.

The DRLLOG DD statement specifies the input IMS log data. You can specify SQNLOGS=1 or leave the default value. If, instead, you are using logs from different IMS in input and you need to open and merge them online, according to a time criteria, then you have to use DRLLOG1 DD, DRLLOG2 DD, DRLLOG3 DD, ...DRLLOG9 DD, specifying one statement for each different IMS working in SQ.

For example, if you have four IMS working in Shared Queue configuration, and you use the online Log merging, you need to 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=....

Chapter 10. Administering the IMS Shared Queue feature 125
Administering the IMS Shared Queue feature

- 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.
Using DRSLSLOGP

DRSLSLOGP is a standalone batch program that you can use to run the IMS Shared Queue feature log procedure in a non-Tivoli Decision Support for OS/390 environment. DRSLSLOGP is the equivalent of running the log procedure under the log collector, except that DB2 is never invoked and DB2 tables are not updated.

DRSLSLOGP can create a large amount of output. Anyway the DRSLSLOGP can create an output reduced in average of the 80% of the initial IMS SLDS. You could use it to precollect IMS SLDSes in your remote centers non-Tivoli Decision Support for OS/390 and after send the produced output to main center.

DRSLSLOGP input and output data sets

DRSLSLOGP has these 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 shared queue configuration, merge all the logs produced by each IMS system that is a member of a shared queue configuration, 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.

**DRLICOMP—output composite records**
The output composite records. The IMS Shared Queue support writes the composite records to the data set associated with the ddname DRLICOMP. The IMS Shared Queue composite record layout is changed and not matching with that of the Tivoli Decision Support for OS/390 IMS feature. Either dummy this ddname, allocate it to a previously created output log procedure checkpoint file, or leave it out.

**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 Shared Queue support can process IMS log data in discrete data set level parts without loss of data. The layout is internal. You can dummy this ddname, allocate it to a previously created output log procedure checkpoint file, or leave it out.

**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. The layout is internal. Do not change it. This data set can be large, for a large IMS system with many secondary transactions. You can dummy this ddname, allocate it to a previously created output log procedure checkpoint file, or leave it out.

**DRLOUT—output messages**
The IMS Shared Queue feature writes messages to this ddname. You can allocate this ddname to SYSOUT, a physical data set, or dummy.
Administering the IMS Shared Queue feature

**DRLDUMP—output error information**

The IMS Shared Queue feature writes error information to this ddname. You can allocate this ddname to SYSOUT, a physical data set, or dummy.

**DRLIRPT2—output from report R2**

The IMS Shared Queue support writes report output to this ddname. You should allocate this ddname to a data set or SYSOUT, according to the requirements of the particular report program. For example, DRLSInn2 writes output to ddname DRLIRPT2, which should have a record length of 220 bytes.

**DRLIPARM—input log procedure parameters**

This ddname points to the parameter data set or contains the in-stream parameters.

The DRLLOG DD statement specifies the input IMS log data. You can specify SQNLOGS=1 or leave the default value. If, instead, you are using logs from different IMS in input and you need to open and merge them online, according to a time criteria, then you have to use DRLLOG1 DD, DRLLOG2 DD, DRLLOG3 DD, ...DRLLOGn DD, specifying one statement for each different IMS working in the shared queue you are using.

For example, if you have four IMS working in Shared Queue configuration, and you use the online Log merging, you need to specify four different DRLLOGx DD statements, one for each IMS, 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.

**Running DRLSLOGP**

DRLSLOGP reads log records from the input IMS log and invokes the IMS log procedure with each of them. It replicates the Tivoli Decision Support for OS/390 log collector functions and maintains the same interface with the log procedure. DRLSLOGP output goes to DRLICOMP and DRLIRPT2.

Two sample jobs that can be used to run DRLSLOGP is in the following figure:
To make the log procedure and the collect process most effective, note these operational considerations when using them:

- Use the TABLEFLUSH parameter sparingly. Acquiring more virtual storage to store pending nodes (increase MAXFREE) can guarantee correct and complete output. Anyway there are some scenarios for which it is very useful to code TABLEFLUSH different from zero. For example Fast Path Output Messages (SLUP) defined with the system definition TERMINAL macro option statement FPACK for which the IMS record x'5936' will be never generated or generated after abnormally long time. This could cause the IMS log collector to have a huge number of nodes allocated/pending and never freed with consequent checkpoint files (when used) that grow indefinitely. In this scenario TABLEFLUSH is only solution to write the incomplete composite records and to free pending nodes.

---

Operational considerations

To make the log procedure and the collect process most effective, note these operational considerations when using them:

- Use the TABLEFLUSH parameter sparingly. Acquiring more virtual storage to store pending nodes (increase MAXFREE) can guarantee correct and complete output. Anyway there are some scenarios for which it is very useful to code TABLEFLUSH different from zero. For example Fast Path Output Messages (SLUP) defined with the system definition TERMINAL macro option statement FPACK for which the IMS record x'5936' will be never generated or generated after abnormally long time. This could cause the IMS log collector to have a huge number of nodes allocated/pending and never freed with consequent checkpoint files (when used) that grow indefinitely. In this scenario TABLEFLUSH is only solution to write the incomplete composite records and to free pending nodes.
If you are collecting data from the IMS systems part of a shared queue configuration, it is strongly recommended that you use the IMS Merge Utility DFSLTMG0, before running the collect job, in order to merge all the logs produced by each IMS system that is member of the same shared queue configuration.

If you need to process IMS SLDS from remote systems in Your TDS/DB2 Central System then you might use the Tivoli Decision Support for OS/390 batch process DRLSLOGP remotely in order to generate the output composite records data set DRLICOMP and transfer it to your central location for Tivoli Decision Support for OS/390 processing (COLLECT) and Tivoli Decision Support for OS/390 DB2 tables filling. You might collect the DRLICOMP data set in a normal product collect process using it in the Collect job card DRLLOG. The advantage of this approach is to transmit a smaller data set in size. In average you will experience a space reduction of about 80% from original IMS SLDS to DRLICOMP data set. For example if your IMS SLDS is 1000 Cyls in size then the DRLICOMP produced will be about 200 Cyls. You can furtherly reduce the DRLICOMP data set size for transmit purpose using TERSE on remote system and UNTERSE on central system. This reduction applies only to the transit time component.

SQNLOGS set to a value other than 1 will add a processing overhead to the collect job. The overall collect time will increase, according to the number of logs, and their time distribution. SQNLOGS =1 will follow the normal procedure flow and the performances will result unchanged.

If SQNLOGS = 1 (set or by default) you must use the DRLLOG DD card. The DRLLOGn DD cards can be used only if SQNLOGS > 2, and must match the SQNLOGS value. For example, if SQNLOGS=3, you must use DRLLOG1, DRLLOG2, DRLLOG3. Also note that if SQNLOGS=n you are not allowed to specify a number of DRLLOG other than n. For example, if SQNLOGS=3, and you specify DRLLOG1, DRLLOG2, DRLLOG3, DRLLOG4, a mismatch error is issued.

If SQNLOGS >1 the messages related to the processed logs are followed by the real log name details processed, if needed.

If SQNLOGS=n, where n is between 1 and 9, in the collect job output you will find an informational message: IEC130I DRLLOGx DD STATEMENT MISSING where x = n + 1.

Running the IMS Light feature

To transfer the IMS logs from remote locations to that where the unique centralized DB2 database is located, the IMS Light feature can be used to reduce the amount of data that needs to be transferred for central processing. By using DRLSLOGP as a standalone procedure in the remote location, the R2 report is produced from the complete IMS log. This R2 report (around 90% reduction in original data size can be expected) can then be transferred to the central location where the final collection processing occurs, instead of transferring the complete IMS log. This reduction applies only to the transit time component.

To use this process, some preliminary work needs to be done.
Setting up Load Library for the IMS Light feature

The Load Modules Library needed to run DRLSLOGP standalone in the remote locations where Tivoli Decision Support for OS/390 is not installed, needs to be built and then redistributed. To build this Load Lib at the central location where Tivoli Decision Support for OS/390 is installed, you can customize and use the sample job DRLJIMSL in the Tivoli Decision Support for OS/390 CNTL library.

//DRLJIMSL JOB (ACCT£), 'IMS LIGHT LIB'
******************************************************************************
//** LICENSED MATERIALS - PROPERTY OF IBM
//**
//** 5695-101 (C) Copyright IBM CORPORATION 2003
//** SEE COPYRIGHT INSTRUCTIONS
//**
******************************************************************************
//**
//** NAME: DRLJIMSL
//** STATUS: TIVOLI DECISION SUPPORT FOR OS/390 1.6.0
//**
//** 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 PARMETERS CAREFULLY (DISP, VOL NAMES IN
** INLIB AND OUTLIB DD STATEMENTS).
//** INLIB CONTAINS THE TDS390 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:
** DRL2I61L,... FOR IMS V6R1M0
** DRL2I71L,... FOR IMS V7R1M0
** DRL2I81L,... FOR IMS V8R1M0
//** - DOUBLE CHECK IN THE JOB OUTPUT THAT ALL THE NEEDED
** LOAD MODULES HAVE BEEN CORRECTLY COPIED.
******************************************************************************
//STEP1 EXEC PGM=IEBCOPY
//SYSIN DD *COPY I=INLIB,O=OUTLIB
SELECT MEMBER=(DRLSLOGP, DRL2LOGJ, DRL2CSQJ, DRLPIO24, DRLPMSG)
SELECT MEMBER=(DRLSI61L, DRLSI612, DRLSI61C)
SELECT MEMBER=(DRLSI71L, DRLSI712, DRLSI71C)
SELECT MEMBER=(DRLSI81L, DRLSI812, DRLSI81C)
/

Statement description

INLIB  The input dataset points to the Tivoli Decision Support for OS/390 SMP/E installation Target or Distribution load library (SDRLLOAD or ADRLLOAD) from which the required load modules are copied.
**Administering the IMS Shared Queue feature**

**OUTLIB**  
The output dataset points to the copy output LOADLIB. It will be used in the remote locations as STEPLIB in the DRLSLOGP job.

If you are interested only to a specific version of IMS, the SELECT MEMBER statements related to the other IMS versions (DRLSIxxL and DRLSIxx2) can be deleted from the job.

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

```plaintext
//DRLILOGP JOB (ACCT£),'DRLSLOGP IMS'
//***************************************************************
//* *
//* LICENSED MATERIALS - PROPERTY OF IBM *
//* *
//* 5695-101 (C) COPYRIGHT IBM CORPORATION 2003 *
//* SEE COPYRIGHT INSTRUCTIONS. *
//* *
//***************************************************************
//* *
//* NAME: DRLILOGP *
//* *
//* STATUS: Tivoli Decision Support for OS390 1.6.0 *
//* *
//* FUNCTION: *
//* RUN THE IMS SQ 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=dr160.IMSLIGHT.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=32766)
//DRLIRPT2 DD DISP=SHR,DSN=... OUTPUT RECORDS FROM R2 REPORT/PROCEDURE
//DRLIPARM DD *
IMSID= -- IMS ID FOR THIS IMS SYSTEM
IMSVER=61 -- THIS IS THE DEFAULT IMS RELEASE TO BE PROCESSED
REPORTS=R2(FF) -- CALL THE R2 REPORT/PROCEDURE
/*
```

**Setting up the Tivoli Decision Support collect for the IMS Light feature**

To run the IMS Light feature in the central location where Tivoli Decision Support for OS/390 is installed, you must have installed the IMS feature completely, including the IMS Light objects) and you might run a collect job against the dataset containing the R2 records (mapped with R2_Light record definition) by using one of the following:

- COLLECT CSQ_V610_COLLECL
- COLLECT CSQ_V710_COLLECL
- COLLECT CSQ_V810_COLLECL
Chapter 11. IMS Shared Queue record definitions

This chapter describes the composite record definitions and sections.

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

Composite record definitions

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

**CSQ_Vnnn_R2**
Composite transaction, after being simplified by the record procedure. This record is created using the collect component.

**CSQ_Vnnn_STxxxx**
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 using the collect component.

Composite record sections

IMS Shared Queue composite records are composed of all fields from the source IMS records useful to populate the tables. The IMS Shared Queue engine allocates an internal IMS control block formatted with the remarkable filed from the original IMS records.

Each section of the composite record has a different name and prefix. Table 7 on page 60 explains the sections of the composite record:

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Prefix</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short description</td>
<td>A descriptive section name.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>The actual name of the section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefix</td>
<td>The one-character prefix for all data subsections and fields defined in the record section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explanation</td>
<td>A detailed explanation of the section and the data grouped in it, including the IMS log record types that compose it and any other notes that apply to the record section.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short description</td>
<td>Name</td>
<td>Prefix</td>
<td>Section contents and explanation</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Origin Unit of Work</td>
<td>OUOW</td>
<td>OW</td>
<td>It is the main key of transaction. It is allocated on arriving</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'01' Message queue insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'03' M.Q.I. destination an SMB</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'03' Message queue enqueue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5901' Aoi exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>This is the anchor section for the PUOW sections</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Unit of Work</td>
<td>PUOW</td>
<td>PW</td>
<td>X'01'/X'03' Message queue insert</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'35' Message queue enqueue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'31' Message queue GU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'36' Message queue dequeue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'37' Message commit/transfer</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'38' Message failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5901' EMH input</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5903' EMH output</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5936' EMH dequeue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5937' Fast Path commit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'5938' Fast Path failed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>All the PUOW sections relative to the same OUOW are chained together</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>in one composite record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>All the records carrying the same Process Unit of Work are chained</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>together at the same PUOW section.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td>PSB</td>
<td>PB</td>
<td>X'08' Program scheduled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>X'07' Program terminated</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>These records are in sequence for the program scheduling that</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>processed the input message.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> One of these records may be absent when an incomplete IMS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>log is processed. This situation is especially likely in the case</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>of WFI regions and IFP regions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Each PSB section has a link with a PUOW section in one composite</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>record.</td>
</tr>
</tbody>
</table>
Chapter 12. IMS Shared Queue data tables and lookup tables

The Tivoli Decision Support for OS/390 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 OS/390 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 chapter describes the data tables and lookup table used by the IMS Shared Queue feature. For an explanation of the naming standard used, see “Naming standard for tables” on page 65. For information about the relationships between tables and between records and tables, see “IMS Shared Queue Collect feature data flow” on page 118.

IMS Shared Queue data tables

The data tables for the IMS Shared Queue component are grouped by subcomponent:

- **Transaction Transit Time subcomponent tables**
  - IMS_TRAN_H, _D, _W
  - IMS_TRAN_QUEUE_Q, _D
  - IMS_SYSTEM_TRAN_H, _D
  - IMS_TRAN_QUEUE_QV, _DV

- **Account and Availability subcomponent tables**
  - IMS_PSB_ACCOUNT_H, _D, _W
  - IMS_AVAIL_RESOURCE
  - IMS_AVAILABILITY_T, _D, _W

- **Statistics subcomponent tables**
  The old checkpoint tables remain unchanged and are used by the IMS Shared Queue component:
  - IMS_CHKPT_POOLS_T
  - IMS_CHKPT_REGION_T
  - IMS_CHKPT_STATS_T
  - IMS_CHKPT_IOSAM_T
  - IMS_CHKPT_VSAM_T

**Transaction Transit Time subcomponent tables and views**

The data tables described in this section are for the Transaction Transit Time subcomponent. These tables store counts of transactions, resources used, and response times by transaction code and user ID. They are used for performance, capacity, and service level tuning and troubleshooting.
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 tables to also identify what users did, how their volumes differed, and their response-time experiences.

The default retention periods are:
- IMS_TRAN_H: 10 days
- IMS_TRAN_D: 45 days
- IMS_TRAN_W: 365 days

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_ID</td>
<td>k CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>ORIGIN_LTERM</td>
<td>k CHAR(8)</td>
<td>The IMS-defined logical name for the terminal used to request the transaction or OTMA Tpipe name.</td>
</tr>
<tr>
<td>DESTINATION_LTERM</td>
<td>k CHAR(8)</td>
<td>The IMS-defined logical name for the terminal used to receive the transaction output. Missing if APPC OTMA.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>k CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td>k CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>PST_ID</td>
<td>k CHAR(2)</td>
<td>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.</td>
</tr>
<tr>
<td>TIME (*)</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH.00.00. This applies only to the _H table.</td>
</tr>
<tr>
<td>DATE</td>
<td>k DATE</td>
<td>The date the activities occurred.</td>
</tr>
<tr>
<td>TRANS_TYPE</td>
<td>k CHAR(8)</td>
<td>Activity type as detailed above.</td>
</tr>
</tbody>
</table>
## Column name | Data type | Description
---|---|---
PERFORMANCE_GROUP | 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 | CHAR(8) | The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.

PROCESS_IMS | CHAR(8) | The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.

PERIOD_NAME | 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.

ROUTING_CODE | CHAR(8) | Code used by EMH to enable transactions to be routed to programs within LBL. From 'X'5901' record.

APPC_MODE_NAME | CHAR(8) | The destination Network ID 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.

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.
## Column name Data type Description

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP_DEDB_NH_PUTGET</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_LOGGED_CI</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_MSDB_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_OVERFLOW_BFR</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>INPUT_CSQ</td>
<td>REAL</td>
<td>The total number of input messages issued by transactions and BMP programs queued through IMS Shared Queue.</td>
</tr>
<tr>
<td>INPUT_LOCAL</td>
<td>REAL</td>
<td>The total number of input messages issued by transactions and BMP programs, not using Shared Queue.</td>
</tr>
<tr>
<td>INPUT_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS system ID. This column is derived from the run time parameter MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>NETWORK_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT_CSQ</td>
<td>REAL</td>
<td>The total number of output messages issued by transactions and BMP programs queued through IMS Shared Queue.</td>
</tr>
<tr>
<td>OUTPUT_LOCAL</td>
<td>REAL</td>
<td>The total number of output messages issued by transactions and BMP programs, not using Shared Queue.</td>
</tr>
<tr>
<td>OUTPUT_SEC</td>
<td>REAL</td>
<td>The total time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.</td>
</tr>
<tr>
<td>OUTPUT_CSQ_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>PGM_CPU_APPROX</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>PGM_SWITCHES</td>
<td>REAL</td>
<td>Number of progra-to-program switches.</td>
</tr>
</tbody>
</table>
### IMS Shared Queue data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGM_SWITCHES_SEC</td>
<td>REAL</td>
<td>It is the time from the x'07' of the Root/Children transaction and the x'08' of the children/root transaction.</td>
</tr>
<tr>
<td>PROCESS_SEC</td>
<td>REAL</td>
<td>The total elapsed time that transactions and BMP programs spent processing in the dependent regions, in seconds.</td>
</tr>
<tr>
<td>RESPONSE_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>RESPONSES</td>
<td>REAL</td>
<td>The total number of responding transactions and BMP programs that sent messages to the originating terminal.</td>
</tr>
<tr>
<td>SQ6_TIME</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>CHAR(8)</td>
<td>The Sysplex system ID. This column is derived from the runtime parameter SYSPLEX_NAME because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>TRANS_PRIOR</td>
<td>CHAR(2)</td>
<td>Message priority.</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>REAL</td>
<td>The total number of IMS transactions for the given interval and unique key combination.</td>
</tr>
<tr>
<td>TRANSIT_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
</tbody>
</table>

(*) Field present only in the IMS_TRANS_H table.
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.

The default retention periods for these tables are:

<table>
<thead>
<tr>
<th>Table</th>
<th>Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_TRAN_QUEUE_Q</td>
<td>10 days</td>
</tr>
<tr>
<td>IMS_TRAN_QUEUE_D</td>
<td>35 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH:15.00. This applies only to the _Q table.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>ORIGIN_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.</td>
</tr>
<tr>
<td>PROCESS_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>PERFORMANCE_GROUP</td>
<td>CHAR(1)</td>
<td>The transaction type based on (DRLMINPUT+DRLMPROC) value ranges. It could be:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F = FAST (0.0 - 0.7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>G = GOOD (0.7 - 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>M = MEDIUM (1.5 - 3.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>L = LOW (3.0 - 10.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S = SLOW (above 10)</td>
</tr>
<tr>
<td>QUEUE_TYPE</td>
<td>CHAR(12)</td>
<td>Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.</td>
</tr>
<tr>
<td>INPUT_MSG</td>
<td>REAL</td>
<td>The total number of input messages processed.</td>
</tr>
<tr>
<td>INPUT_MSG_SEC</td>
<td>REAL</td>
<td>The total time, in seconds, the input messages spent on that queue.</td>
</tr>
<tr>
<td>INP_EMHQ_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the input queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>INP_EMHQ_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the input queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>INP_EMHQ_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the input queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>INP_MSGQ_SHMSG_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the input short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>INP_MSGQ_SHMSG_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the input short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>INP_MSGQ_SHMSG_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the input short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>INP_MSGQ_LMSG_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the input long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>INP_MSGQ_LMSG_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the input long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>INP_MSGQ_LMSG_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the input long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>MVS system ID.</td>
</tr>
<tr>
<td>OUTPUT_MSG</td>
<td>REAL</td>
<td>The total number of output messages processed.</td>
</tr>
<tr>
<td>OUTPUT_MSG_SEC</td>
<td>REAL</td>
<td>The total time, in seconds, the output messages spent on that queue.</td>
</tr>
<tr>
<td>OUT_EMHQ_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the output queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>OUT_EMHQ_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the output queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>OUT_EMHQ_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the output queue for fast path transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_SHMSG_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the output short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_SHMSG_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the output short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_SHMSG_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the output short message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_LMSG_AVG(*)</td>
<td>REAL</td>
<td>Average number of the messages on the output long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_LMSG_MAX(*)</td>
<td>REAL</td>
<td>Maximum number of the messages on the output long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>OUT_MSGQ_LMSG_MIN(*)</td>
<td>REAL</td>
<td>Minimum number of the messages on the output long message queue for full function transactions present before processing.</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>CHAR(8)</td>
<td>Sysplex name.</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>REAL</td>
<td>The total number of IMS transactions for the given interval and unique key.</td>
</tr>
</tbody>
</table>

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

1. 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.

2. 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.
IMS Shared Queue data tables and lookup tables

3. 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.

4. In the Shared Queue environment, these fields contain valid data only if the collection is done from a merged log.
**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.

The default retention periods for these tables are:

<table>
<thead>
<tr>
<th>Table</th>
<th>Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_SYSTEM_TRAN_H</td>
<td>10 days</td>
</tr>
<tr>
<td>IMS_SYSTEM_TRAN_D</td>
<td>45 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>PST_ID</td>
<td>CHAR(2)</td>
<td>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.</td>
</tr>
<tr>
<td>TIME (*)</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.00.00. This applies only to the _H table.</td>
</tr>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>The date the activities occurred.</td>
</tr>
<tr>
<td>ORIGIN_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.</td>
</tr>
<tr>
<td>PROCESS_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>FF_ABORTS</td>
<td>REAL</td>
<td>The total number of Full Function transactions that aborted their commits.</td>
</tr>
<tr>
<td>FF_COMMITS</td>
<td>REAL</td>
<td>The total number of Full function transactions that completed their commits.</td>
</tr>
<tr>
<td>FP_ABORTS</td>
<td>REAL</td>
<td>The total number of Fast Path transactions that aborted their commits.</td>
</tr>
<tr>
<td>FP_CI_NHNH_CONT</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FP_COMBINATIONS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_COMMITS</td>
<td>REAL</td>
<td>The total number of Fast Path transactions that completed their commits.</td>
</tr>
<tr>
<td>FP_DEDB_BFR_WAITS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_DEDB_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_DEDB_NH_PUTGET</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_LOGGED_CI</td>
<td>REAL</td>
<td>The total number of Fast Path whole IDs logged, derived from the count stored in the FP syncpoint record (record type X'5937' or X'5938'). This represents the number of whole IDs logged by the program while the transactions were active.</td>
</tr>
<tr>
<td>FP_MSDB_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>FP_OVERFLOW_BFR</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>INPUT_CSQ</td>
<td>REAL</td>
<td>The total number of input messages issued by transactions and BMP programs queued through IMS Shared Queue.</td>
</tr>
<tr>
<td>INPUT_LOCAL</td>
<td>REAL</td>
<td>The total number of input messages issued by transactions and BMP programs, not using Shared Queue.</td>
</tr>
<tr>
<td>INPUT_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>The MVS system ID. This column is derived from the run time parameter MVS_SYSTEM_ID because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>NETWORK_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>OUTPUT_CSQ</td>
<td>REAL</td>
<td>The total number of output messages issued by transactions and BMP programs queued through IMS Shared Queue.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OUTPUT _LOCAL</td>
<td>REAL</td>
<td>The total number of output messages issued by transactions and BMP programs, not using Shared Queue.</td>
</tr>
<tr>
<td>OUTPUT _SEC</td>
<td>REAL</td>
<td>The total time that responding transactions spent on the IMS output queue waiting for transmission to the ultimate network destination, in seconds.</td>
</tr>
<tr>
<td>OUTPUT_CSQ_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>PGM_SWITCHES</td>
<td>REAL</td>
<td>Number of program-to-program switches.</td>
</tr>
<tr>
<td>PGM_SWITCHES_SEC</td>
<td>REAL</td>
<td>It is the time from the x’07’ of the Root/Children transaction and the x’08’ of the children/root transaction.</td>
</tr>
<tr>
<td>PROCESS_SEC</td>
<td>REAL</td>
<td>The total elapsed time that transactions and BMP programs spent processing in the dependent regions, in seconds.</td>
</tr>
<tr>
<td>RESPONSE_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>RESPONSES</td>
<td>REAL</td>
<td>The total number of responding transactions and BMP programs that sent messages to the originating terminal.</td>
</tr>
<tr>
<td>SQ6_TIME</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>CHAR(8)</td>
<td>The Sysplex system ID. This column is derived from the run time parameter SYSPLEX_NAME because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>TRAN_CNTR_1</td>
<td>REAL</td>
<td>The total number of IMS transactions whose transit time was less than the user-specified boundary 1 (default for boundary is 0.7 seconds).</td>
</tr>
<tr>
<td>TRAN_CNTR_2</td>
<td>REAL</td>
<td>The total number of IMS transactions whose transit time was less than the user-specified boundary 2 (default for boundary is 1.5 seconds).</td>
</tr>
<tr>
<td>TRAN_CNTR_3</td>
<td>REAL</td>
<td>The total number of IMS transactions whose transit time was less than the user-specified boundary 3 (default for boundary is 3 seconds).</td>
</tr>
<tr>
<td>TRAN_CNTR_4</td>
<td>REAL</td>
<td>The total number of IMS transactions whose transit time was less than the user-specified boundary 4 (default for boundary is 10 seconds).</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>REAL</td>
<td>The total number of IMS transactions for the given interval and unique key combination.</td>
</tr>
<tr>
<td>TRANSIT_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
</tbody>
</table>

(*) Field present only in the IMS_SYSTEM_TRAN_H table.

**TRANS_TYPE key column:** The TRA3N_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 Shared Queue
- L Local Shared Queue
- 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
- 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
IMS Shared Queue data tables and lookup tables

C         Input cancelled
A         Aborted, or program abended
-         Not available

Views

**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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date when the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>TIME</td>
<td>The time when the activity started, in the format HH.15.00. This applies only to the _Q table.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>ORIGIN_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin.</td>
</tr>
<tr>
<td>PROCESS_IMS</td>
<td>CHAR(8)</td>
<td>The IMS subsystem ID defined in the processing part of the UOW token. It identifies the activity processor.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>PERFORMANCE_GROUP</td>
<td>CHAR(1)</td>
<td>The transaction type based on (DRLMINPUT+ DRLMPROCE) value ranges. It could be:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ \begin{align*} F &amp;= \text{FAST} &amp; (0.0 - 0.7) \ G &amp;= \text{GOOD} &amp; (0.7 - 1.5) \ M &amp;= \text{MEDIUM} &amp; (1.5 - 3.0) \ L &amp;= \text{LOW} &amp; (3.0 - 10.0) \ S &amp;= \text{SLOW} &amp; (\text{above 10}) \end{align*} ]</td>
</tr>
<tr>
<td>QUEUE_TYPE</td>
<td>CHAR(12)</td>
<td>Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.</td>
</tr>
<tr>
<td>INPUT_MSG_SEC_AVG</td>
<td>REAL</td>
<td>Average time spent by an input message on that queue. Calculated as INPUT_MSG_SEC / INPUT_MSG.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>MVS system ID.</td>
</tr>
<tr>
<td>OUTPUT_MSG_SEC_AVG</td>
<td>REAL</td>
<td>Average time spent by an output message on that queue. Calculated as OUTPUT_MSG_SEC / OUTPUT_MSG.</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>CHAR(8)</td>
<td>Sysplex name.</td>
</tr>
<tr>
<td>TRANS_RATE</td>
<td>REAL</td>
<td>Average transaction arrival rate over the interval. Calculated as TRANSACTIONS / 900 (for quarter-hourly) or TRANSACTIONS / 86400 (for daily).</td>
</tr>
</tbody>
</table>

**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 140.

Chapter 12. IMS Shared Queue data tables and lookup tables 147
Account and Availability subcomponent tables

The data tables described in this section are for the Account and Availability subcomponent. These tables provide resource consumption and availability data about the IMS systems, regions, and applications.

Data Tables

**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 transaction scheduling a PSB. The PSB Account tables give statistics for CPU time and elapsed time during a specified period for Regions, Transactions, Programs (PSB). This 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 DB operations.

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

The default retention periods for these tables are:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Default Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_PSB_ACCOUNT_H</td>
<td>10 days</td>
</tr>
<tr>
<td>IMS_PSB_ACCOUNT_D</td>
<td>45 days</td>
</tr>
<tr>
<td>IMS_PSB_ACCOUNT_W</td>
<td>365 days</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>k DATE</td>
<td>The date the activities occurred. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>TIME</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH:00:00. This applies only to the _H table.</td>
</tr>
<tr>
<td>IMS_SUBSYSTEM_NAME</td>
<td>k CHAR(8)</td>
<td>The IMS subsystem name. From DLRTOKN.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k CHAR(8)</td>
<td>The name of the IMS transaction the user requested. From DLRTCN.</td>
</tr>
<tr>
<td>PSB_NAME</td>
<td>k CHAR(8)</td>
<td>The name of the IMS program used to process the transaction. This column contains the program specification block (PSB). From DLRTNS.</td>
</tr>
<tr>
<td>PST_ID</td>
<td>k CHAR(2)</td>
<td>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.</td>
</tr>
<tr>
<td>REGION_JOB_NAME</td>
<td>k CHAR(8)</td>
<td>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.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>APPLICATION_NAME</td>
<td>CHAR(18)</td>
<td>Application name. This is from the RESOURCE_TARGET_NM in IMS_AVAILRESOURCE lookup table. If nothing is found, $UNKNOWN is used as default.</td>
</tr>
<tr>
<td>DLI_APSB_CALLS</td>
<td>REAL</td>
<td>Number of DL/I APSB calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAPSB.</td>
</tr>
<tr>
<td>DLI_CHKP_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_CMD_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_DPSB_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_EXCL_DEQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_EXCL_ENQUEUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_EXCL_ENQWAITS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_GCMD_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_GMSG_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_ICMD_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_INIT_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_INQV_CALLS</td>
<td>REAL</td>
<td>Number of DL/I INQV calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRINQV.</td>
</tr>
<tr>
<td>DLI_MSG_AUTH_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_MSG_CHNG_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DLI_MSG_SETO_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_PURGE_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_QCMD_DEQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_QCMD_ENQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_QCMD_ENQWAITS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_RCMD_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_ROLB_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_ROLS_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_SETS_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_SETU_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_SLOG_CALLS</td>
<td>REAL</td>
<td>Number of DL/I SLOG calls, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRSLOG.</td>
</tr>
<tr>
<td>DLI_TEST_DEQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_TEST_ENQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_TEST_ENQWAITS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_UPDT_DEQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_UPDT_ENQUEUES</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLI_UPDT_ENQWAITS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DLI_XRST_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDB_CALLS</td>
<td>REAL</td>
<td>The total number of DL/I database calls issued, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRDELT.</td>
</tr>
<tr>
<td>DLIDB_DLET_CALLS</td>
<td>REAL</td>
<td>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 DLRGHNP.</td>
</tr>
<tr>
<td>DLIDB_GHN_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDB_GHNP_CALLS</td>
<td>REAL</td>
<td>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 DLRGHU.</td>
</tr>
<tr>
<td>DLIDB_GU_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDB_ISRT_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDB_REPL_CALLS</td>
<td>REAL</td>
<td>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 DLRTMEPL.</td>
</tr>
<tr>
<td>DLIDB_CTL_DBIO_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDB_CTL_DBIOS</td>
<td>REAL</td>
<td>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 DLRTMEPL.</td>
</tr>
<tr>
<td>DLIDB_CTL_LOCK_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDC_GN_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDC_GU_CALLS</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLIDC_ISRT_CALLS</td>
<td>REAL</td>
<td>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 DLRISRT.</td>
</tr>
</tbody>
</table>
### IMS Shared Queue data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLIACC_ACCUM_SEC</td>
<td>REAL</td>
<td>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.</td>
</tr>
<tr>
<td>DLISQ6_ACCUM_SEC</td>
<td>REAL</td>
<td>The Region Subq 6 Time per Message in seconds. Calculated as Sum of DLRSQ6TM.</td>
</tr>
<tr>
<td>LAST_MSG_USERID</td>
<td>CHAR(8)</td>
<td>User ID of the last message processed in this dependent region. From DLRUSID.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>CHAR(4)</td>
<td>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 because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>PGM_CPU_SEC</td>
<td>REAL</td>
<td>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</td>
</tr>
<tr>
<td>PGM_FAILURES</td>
<td>REAL</td>
<td>The total number of program abends.</td>
</tr>
<tr>
<td>PROGRAM_TYPE</td>
<td>Char(10)</td>
<td>From DRLTYPE. When:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'80'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'10'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'08'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'04'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'02'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'01'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'22'</td>
</tr>
<tr>
<td></td>
<td></td>
<td>X'21'</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>CHAR(8)</td>
<td>This column is derived from the run time parameter :SYSPLEX_NAME because IMS log records do not contain this field.</td>
</tr>
<tr>
<td>TRANS_PRIOR</td>
<td>CHAR(2)</td>
<td>Transaction priority. From DLRPRTY.</td>
</tr>
<tr>
<td>TRANSACTIONS</td>
<td>REAL</td>
<td>Number of transactions processed by the PSB. Calculated as Sum of DLRMCNT</td>
</tr>
</tbody>
</table>
Availability for IMS resources

Availability for IMS resources is relative to the following types of resource:

- IMS subsystem
- IMS region

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

For the IMS subsystem, the following IMS records are used to track availability:

- Record 06: IMS subsystem Start/Stop
- Record 4001: IMS system checkpoint

For the IMS region, the following IMS record is used to track availability:

- Record 47: IMS region checkpoint

The following tables are used:

- IMS_AVAIL_RESOURCE: this lookup table defines which IMS resources are used for tracking availability. These values are used during the update of 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.
- IMS_AVAILABILITY_T
- IMS_AVAILABILITY_D,_W
## 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_AVAILRESOURCE lookup table.

The default retention period for this table is 10 days.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPLEX_NAME</td>
<td>k Char(8)</td>
<td>Sysplex name. This is from the SYSPLEX_NAME collect parameter.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k Char(4)</td>
<td>MVS system ID. This is from the MVS_SYSTEM_ID collect parameter.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k Char(8)</td>
<td>The IMS subsystem ID. This is from the IMS_SYSTEM_ID collect parameter or from the specific field for the different record type collected.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>k Char(18)</td>
<td>Resource Name. This is from the RESOURCE_TARGET_NM or from RESOURCE_SOURCE_NM into IMS_AVAILRESOURCE lookup table.</td>
</tr>
<tr>
<td>RESOURCE_TYPE</td>
<td>k Char(8)</td>
<td>Resource Type. Possible values are: IMSSYS IMS system, IMSREG IMS region.</td>
</tr>
<tr>
<td>INTERVAL_TYPE</td>
<td>k Char(3)</td>
<td>Interval type. Possible values are: ====, ==</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>START_TIME</td>
<td>k TIMESTAMP</td>
<td>Start time of the interval.</td>
</tr>
<tr>
<td>END_TIME</td>
<td>TIMESTAMP</td>
<td>End time of the interval.</td>
</tr>
<tr>
<td>QUIET_INTERVAL_SEC</td>
<td>INTEGER</td>
<td>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.</td>
</tr>
</tbody>
</table>
IMS Availability_D, W

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

The default retention period for these tables are:
- IMS_AVAILABILITY_D: 45 days
- IMS_AVAILABILITY_W: 365 days

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DATE</td>
<td>Date the availability data applies to. For the _W table, this is the date of the first day of the week.</td>
</tr>
<tr>
<td>SYSPLEX_NAME</td>
<td>Char(8)</td>
<td>Sysplex name.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>Char(4)</td>
<td>MVS system ID.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>Char(8)</td>
<td>The IMS subsystem ID.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>Char(18)</td>
<td>Resource Name.</td>
</tr>
<tr>
<td>RESOURCE_TYPE</td>
<td>Char(8)</td>
<td>Resource Type. Possible values are: IMSSYS (IMS system) IMSREG (IMS region)</td>
</tr>
<tr>
<td>AVAIL_OBJ_PCT</td>
<td>DECIMAL (4,1)</td>
<td>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</td>
</tr>
<tr>
<td>MEASURED_HOURS</td>
<td>FLOAT</td>
<td>Number of hours measured.</td>
</tr>
<tr>
<td>SCHEDULE_HOURS</td>
<td>FLOAT</td>
<td>Number of hours the resource is scheduled to be up.</td>
</tr>
<tr>
<td>STARTS</td>
<td>SMALLINT</td>
<td>Number of times the resource was started.</td>
</tr>
<tr>
<td>STARTS_IN_SCHEDULE</td>
<td>SMALLINT</td>
<td>Number of times the resource was started within the schedule.</td>
</tr>
<tr>
<td>STOPS</td>
<td>SMALLINT</td>
<td>Number of times the resource was stopped.</td>
</tr>
<tr>
<td>STOPS_IN_SCHEDULE</td>
<td>SMALLINT</td>
<td>Number of times the resource was stopped within the schedule.</td>
</tr>
<tr>
<td>UP_HOURS</td>
<td>FLOAT</td>
<td>Number of hours the resource was up.</td>
</tr>
<tr>
<td>UP_IN_SCHEDULE</td>
<td>FLOAT</td>
<td>Number of hours the resource was up within the schedule.</td>
</tr>
</tbody>
</table>

Lookup tables

First Paragraph

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.

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSPLEX_NAME</td>
<td>Char(8)</td>
<td>Sysplex name that the resource is associated with. This can contain global search characters.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>Char(4)</td>
<td>MVS system ID that the resource is associated with. This can contain global search characters.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>Char(8)</td>
<td>IMS subsystem ID that the resource is associated with. This can contain global search characters.</td>
</tr>
</tbody>
</table>
IMS Shared Queue data tables and lookup tables

<table>
<thead>
<tr>
<th>Column name</th>
<th>Data type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOURCE_SOURCE_NM</td>
<td>Char(8)</td>
<td>Resource source name. This specifies the resource you want to monitor. This can contain global search characters.</td>
</tr>
</tbody>
</table>
| RESOURCE_TYPE        | Char(8)   | Resource type:  
| IMSSYS               | IMS system                                                                                                                                     |
| IMSREG               | IIMS 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. |
| SCHEDULE_NAME        | Char(8)   | Schedule name to use for the resource. If nothing is specified, STANDARD is used as default.                                                                                                           |

Example of Table Contents:

<table>
<thead>
<tr>
<th>SYSPLEX MVS SYSTEM ID</th>
<th>IMS SYSTEM ID</th>
<th>RESOURCE SOURCE NM</th>
<th>RESOURCE TYPE</th>
<th>RESOURCE TARGET NM</th>
<th>RESOURCE CHKPOINT</th>
<th>SCHEDULE NAME</th>
<th>AVAIL OBJ PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>%</td>
<td>IMS71A</td>
<td>IMSSYS</td>
<td>---</td>
<td>3600</td>
<td>STANDARD</td>
<td>95.0</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>DSWM</td>
<td>IMSREG</td>
<td>---</td>
<td>3600</td>
<td>STANDARD</td>
<td>95.0</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>MPP</td>
<td>IMSREG</td>
<td>MPP</td>
<td>3600</td>
<td>STANDARD</td>
<td>95.0</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>PGM</td>
<td>IMSPGM</td>
<td>APPL1</td>
<td>---</td>
<td>STANDARD</td>
<td>95.0</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>TRAN</td>
<td>IMSTRAN</td>
<td>APPL1</td>
<td>---</td>
<td>STANDARD</td>
<td>95.0</td>
</tr>
</tbody>
</table>
Part 3. Appendixes
Appendix A. Reports

Tivoli Decision Support for OS/390 produces reports based on the data in the Tivoli Decision Support for OS/390 database. Reports can show data from tables or from views. You can request reports using the Tivoli Decision Support for OS/390 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 chapter describes the reports provided with the collect component of the IMS Performance 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 OS/390 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 Performance 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.
- A one-character identifier of the IMS Performance feature subcomponent that provides the report:
  - T: Transaction subcomponent report
  - Y: System subcomponent report
  - A: Application subcomponent report
  - S: Statistics 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 OS/390 organizes reports into report groups, which correspond to feature components. IMS Performance feature reports belong to the IMS report group.

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 control task
  - Service level planning task
  - Capacity planning task
  - Security control task
  - Configuration management discipline
  - Operations management discipline
  - Change management discipline
  - 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 OS/390 prompts you for the variables listed in the description.

Report types

The IMS Performance 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 28 shows an example of a tabular report.

<table>
<thead>
<tr>
<th>Period</th>
<th>Model</th>
<th>LCU number</th>
<th>Device number</th>
<th>Volser</th>
<th>I/O Response rate /sec</th>
<th>Response avg mpi</th>
<th>Response avg msec</th>
<th>Queue avg msec</th>
<th>Connect avg msec</th>
<th>Disconnect avg msec</th>
<th>Pending avg msec</th>
<th>Allocs avg</th>
<th>Dev busy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>014E</td>
<td>DBSL05</td>
<td>9.1</td>
<td>0.076</td>
<td>11.3</td>
<td>3.3</td>
<td>3.1</td>
<td>0.1</td>
<td>4.8</td>
<td>119.4</td>
<td>5.44</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>09</td>
<td>0056</td>
<td>COR105</td>
<td>5.4</td>
<td>0.036</td>
<td>6.1</td>
<td>0.3</td>
<td>3.2</td>
<td>0.1</td>
<td>2.5</td>
<td>122.8</td>
<td>3.41</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>0142</td>
<td>TSOL01</td>
<td>4.7</td>
<td>0.015</td>
<td>6.8</td>
<td>0.4</td>
<td>2.6</td>
<td>0.1</td>
<td>3.7</td>
<td>176.8</td>
<td>3.19</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>09</td>
<td>0122</td>
<td>SF2053</td>
<td>1.1</td>
<td>0.033</td>
<td>27.6</td>
<td>1.6</td>
<td>12.5</td>
<td>0.4</td>
<td>13.1</td>
<td>55.6</td>
<td>2.90</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>0148</td>
<td>TSOL05</td>
<td>2.6</td>
<td>0.031</td>
<td>9.7</td>
<td>2.3</td>
<td>2.4</td>
<td>0.2</td>
<td>4.8</td>
<td>74.8</td>
<td>2.08</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>0143</td>
<td>TSOL04</td>
<td>8.2</td>
<td>0.029</td>
<td>4.6</td>
<td>0.4</td>
<td>2.2</td>
<td>0.3</td>
<td>1.6</td>
<td>154.4</td>
<td>2.63</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>09</td>
<td>0106</td>
<td>COR106</td>
<td>1.2</td>
<td>0.025</td>
<td>20.4</td>
<td>0.9</td>
<td>3.6</td>
<td>0.3</td>
<td>15.6</td>
<td>826.6</td>
<td>2.33</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>0141</td>
<td>TSOL02</td>
<td>3.2</td>
<td>0.023</td>
<td>6.6</td>
<td>0.2</td>
<td>2.7</td>
<td>0.1</td>
<td>3.7</td>
<td>120.2</td>
<td>2.19</td>
<td></td>
</tr>
<tr>
<td>PRIME</td>
<td>0F</td>
<td>0148</td>
<td>ICR003</td>
<td>9.7</td>
<td>0.021</td>
<td>2.7</td>
<td>0.1</td>
<td>2.2</td>
<td>0.0</td>
<td>0.3</td>
<td>2.9</td>
<td>1.91</td>
<td></td>
</tr>
</tbody>
</table>

Tivoli Decision Support for OS/390 Report: MVSPM05

Figure 28. 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 OS/390 have both a QMF format and a Graphical Data Display Manager (GDDM®) format. Figure 29 illustrates a graphic report.

Figure 29. 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-SQ and SQ tables**

If you decide to use:

- Tivoli Decision Support for OS/390 shared queue engine for IMS 6.1 and 7.1 not running in shared queue
- Tivoli Decision Support for OS/390 shared queue engine for IMS 6.1 and 7.1 running in shared queue
- Tivoli Decision Support for OS/390 shared queue engine for IMS 8.1 running in shared queue or in non-shared queue

you will have to build new reports, based on the new provided ones. The reason for this is that the IMS Shared Queue support uses tables that have different fields and structure from the pre-Shared Queue support.

Also, if you need to report data across the Shared Queue time installation boundary (that is, some data from the previous pre-Shared Queue tables and some data from the new Shared Queue tables), you might need to build new reports that will select data from both the old and the new tables.

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 30 on page 163.
This query uses the following fields from the IMS_SYSTEM_D pre-SQ 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 Shared Queue implementation engine in the product, you need to use a different query, but with a similar structure to that shown in the Figure 31 on page 164.
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 31. Example of Query from IMS_SYSTEM_TRAN_D Shared Queue Table

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

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

Now, if you want to generate a unique report including data from both the pre-SQ table and the new SQ table you can build a new query using the UNION SQL keyword.

The query might look like this:
In Figure 32 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-Shared Queue 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 Shared Queue environment tables.

For example, assuming that you want to run this report for the whole month of January 2003, but until January, 14th you used the old Tivoli Decision Support for

```sql
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(TXNS)), 0)*100
  , VALUE(
    (SUM(TRAN_CNTR_2))
     /(SUM(TXNS)), 0)*100
  , VALUE(
    (SUM(TRAN_CNTR_3))
     /(SUM(TXNS)), 0)*100
  , VALUE(
    (SUM(TRAN_CNTR_4))
     /(SUM(TXNS)), 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 32. Example of Mixed Query
OS/390 engine, and from January, 15th you started using the new Tivoli Decision
Support for OS/390 SQ engine, when you are prompted for the input variables in
the data selection panel (DRLDSEL), you can select the correct date range, as
follows:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>ZOS1</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>IMS1</td>
</tr>
<tr>
<td>PRESQ_FROM_DATE</td>
<td>2003-01-01</td>
</tr>
<tr>
<td>PRESQ_TO_DATE</td>
<td>2003-01-14</td>
</tr>
<tr>
<td>NEWSQ_FROM_DATE</td>
<td>2003-01-15</td>
</tr>
<tr>
<td>NEWSQ_TO_DATE</td>
<td>2003-01-31</td>
</tr>
</tbody>
</table>

Figure 33. Example of Query Input Variables Panel DRLDSEL

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 overview reports

IMS overview reports reflect an overview of items for a specified time period.

**IMS Application Response Time Overview report**

For the IMS applications selected, this report shows the percentage of response times within each boundary or threshold, for a selected week. You select the week by specifying any day in the week. Since the data for this report is retained in the Tivoli Decision Support for OS/390 database for 70 days by default, the report may contain days with no data shown. Figure 34 illustrates a overview report.

This information identifies the report:

- **Report ID**: IMSA04
- **Report group**: IMS reports
- **Source**: IMS_APPLICATION_H
- **Attributes**: IMS, performance, response, application, overview
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, DATE_IN_WEEK

![Figure 34. Example of IMS Application Response Time Overview graphic report](image)

The report contains this information:

- **Day**: Day of the week the data is for
- **Vertical axis**: Percent of transactions
Reports

Percentage transactions within 1
Percent of transactions with transit time less than response-time boundary 1 (default is 1 second).

Percentage transactions within 2
Percent of transactions with transit time less than response-time boundary 2 (default is 2 seconds).

Percentage transactions within 3
Percent of transactions with transit time less than response-time boundary 3 (default is 5 seconds).

Percentage transactions within 4
Percent of transactions with transit time less than response-time boundary 4 (default is 10 seconds).

Percentage transactions above 4
Percent of transactions with transit time equal to or greater than response-time boundary 4 (default is 10 seconds).
IMS Application Transaction Overview report

For IMS applications selected, this report indicates the number of transactions processed within each response time boundary or threshold, for a selected week. You select the week by specifying any day in the week. Since the data for this report is retained in the Tivoli Decision Support for OS/390 data base for 70 days by default, the report may contain days with no data shown. Figure 35 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSA05
- **Report group**: IMS reports
- **Source**: IMS_APPLICATION_H
- **Attributes**: IMS, performance, volumes, application, overview
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, DATE_IN_WEEK

The report contains this information:

- **Day**: Day of the week the data is for

**Vertical axis**: Number of transactions

**Transactions within boundary 1**: Number of transactions with transit time less than response-time boundary 1 (default is 1 second).

![Figure 35. Example of IMS Application Transaction Overview graphic report](image_url)
Reports

Transactions within boundary 2
Number of transactions with transit time less than response-time boundary 2 (default is 2 seconds).

Transactions within boundary 3
Number of transactions with transit time less than response-time boundary 3 (default is 5 seconds).

Transactions within boundary 4
Number of transactions with transit time less than response-time boundary 4 (default is 10 seconds).

Total number of transactions
Total number of transactions for the selected applications and day.
IMS trend reports

IMS trend reports show you trends over a specified time period.

**IMS Application Response Time Trend report**

For the application you select, this report shows the percentage of response times within each boundary or threshold by week between the FROM_DATE and TO_DATE specified. Figure 36 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSA01
- **Report group**: IMS reports
- **Source**: IMS_APPLICATION_W
- **Attributes**: IMS, performance, response, application, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, FROM_DATE, TO_DATE

![Figure 36. Example of IMS Application Response Time Trend graphic report](image)

The report contains this information:

- **Week start date**: Date of the first day in the week the data is for
- **Vertical axis**: Percent of transactions
Response time boundary 1
Percent of transactions with transit time less than response-time boundary 1 (default is 1 second).

Response time boundary 2
Percent of transactions with transit time less than response-time boundary 2 (default is 2 seconds).

Response time boundary 3
Percent of transactions with transit time less than response-time boundary 3 (default is 5 seconds).

Response time boundary 4
Percent of transactions with transit time less than response-time boundary 4 (default is 10 seconds).
IMS Application Transaction Trend report

For each application you select, this report shows the number of transactions processed by week between the FROM_DATE and TO_DATE specified. Figure 37 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSA02
- **Report group**: IMS reports
- **Source**: IMS_APPLICATION_W
- **Attributes**: IMS, performance, volumes, application, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, FROM_DATE, TO_DATE

![Figure 37. Example of IMS Application Transaction Trend graphic report](image)

The report contains this information:

- **Week start date**: Date of the first day in the week the data is for
- **Vertical axis**: Number of transactions
- **Message queue responses**: The total number of responding message-driven transactions and BMP programs that sent messages to the original terminal.
- **EMH transactions**: The total number of IMS Fast Path transactions
IMS Application CPU Utilization Trend report

This report shows the CPU resource consumed by a selected application by week between the FROM_DATE and TO_DATE specified. Figure 38 shows an example of a report.

This information identifies the report:

<table>
<thead>
<tr>
<th>Report ID</th>
<th>IMSA03</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report group</td>
<td>IMS reports</td>
</tr>
<tr>
<td>Source</td>
<td>IMS_APPLICATION_W</td>
</tr>
<tr>
<td>Attributes</td>
<td>IMS, performance, utilization, application, trend</td>
</tr>
<tr>
<td>Variables</td>
<td>MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, FROM_DATE, TO_DATE</td>
</tr>
</tbody>
</table>

![Figure 38. Example of IMS Application CPU Utilization Trend graphic report](image)

The report contains this information:

**Week start date**
- Date of the first day in the week the data is for

**Vertical axis**
- Seconds

**Program CPU time (seconds)**
- The total dependent region CPU TCB 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).
- This is the sum of the approximate number of CPU seconds of program execution time while the transactions were active.
IMS System Response Time Trend report

For the IMS system you select, this report shows the percentage of response times within each boundary or threshold by week between the FROM_DATE and TO_DATE specified. Figure 39 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY01
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_D
- **Attributes**: IMS, performance, system, transaction, daily, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, FROM_DATE, TO_DATE

![IMS System Response Time Trend graphic report](image)

*Figure 39. Example of IMS System Response Time Trend graphic report*

The report contains this information:

- **Week start date**
  - First day of the week the data is for

- **Vertical axis**
  - Percent of transactions

- **Percentage transactions within 1**
  - Percent of transactions with transit time less than response-time boundary 1 (default is 1 second).

- **Percentage transactions within 2**
  - Percent of transactions with transit time less than response-time boundary 2 (default is 2 seconds).
Percentage transactions within 3
Percent of transactions with transit time less than response-time boundary 3 (default is 5 seconds).

Percentage transactions within 4
Percent of transactions with transit time less than response-time boundary 4 (default is 10 seconds).
IMS System Transaction Volumes Trend report

For the IMS system you select, this report indicates the volume of transactions processed and completed within each response time boundary by week between the FROM_DATE and TO_DATE specified. Figure 40 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY02
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_D
- **Attributes**: IMS, performance, volumes, system, daily, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, FROM_DATE, TO_DATE

![IMS System Transaction Volumes Trend graphic report](image)

**Figure 40. Example of IMS System Transaction Volumes Trend graphic report**

The report contains this information:

- **Week start date**: First day of the week the data is for
- **Vertical axis**: Number of transactions
- **Transactions within 1**: Number of transactions with transit time less than response-time boundary 1 (default is 1 second).
- **Transactions within 2**: Number of transactions with transit time less than response-time boundary 2 (default is 2 seconds).
**Reports**

**Transactions within 3**
Number of transactions with transit time less than response-time boundary 3 (default is 5 seconds).

**Transactions within 4**
Number of transactions with transit time less than response-time boundary 4 (default is 10 seconds).

**Total transactions**
Total number of transactions for the selected IMS system.
IMS System CPU Utilization Trend report

This report shows the CPU resource consumed by an IMS system by week between the FROM_DATE and TO_DATE specified. Figure 41 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY03
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_D
- **Attributes**: IMS, performance, utilization, system, daily, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, FROM_DATE, TO_DATE

Figure 41. Example of IMS System CPU Utilization Trend graphic report

The report contains this information:

- **Week start date**: First day of the week the data is for
- **Vertical axis**: CPU time in seconds
- **Program CPU time (seconds)**: The total dependent region CPU TCB 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).

This represents the sum of the approximate number of CPU seconds of program execution time while transactions were active.
IMS System DLI Utilization Trend report

This report indicates the DL/I call counts issued by an IMS system by week between the FROM_DATE and TO_DATE specified. Figure 42 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY04
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_D
- **Attributes**: IMS, performance, utilization, system, daily, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, FROM_DATE, TO_DATE

---

![Figure 42. Example of IMS System DLI Utilization Trend graphic report](image)

The report contains this information:

- **Week start date**: First day of the week the data is for
- **Vertical axis**: Number of database calls
- **DLI database calls**: The total number of DL/I database calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate total number of DL/I database calls that the program issued while the transactions were active.
IMS detail reports

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

**IMS User ID Response Time and CPU Detail by Date report**

This report shows the resources used by the users logged onto IMS during the day selected. Figure 43 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSA06
- **Report group**: IMS reports
- **Source**: IMS_USER_APPL_D
- **Attributes**: IMS, user ID, utilization, performance, response, daily
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, USER_ID, DATE

The report contains this information:

**User ID**

The user ID or the logical terminal name.

**EMH response time**

The sum of the total end-to-end user perceived response time, in seconds. This normally includes user think time and therefore cannot be used easily to gauge true end-user response times.

**MSGQ response time**

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.

**Average EMH response time**

The average user-perceived response time, in seconds.

**Average MSGQ response time**

The average network transmission time for transactions, in seconds.

**Program CPU time**

The total dependent region CPU TCB 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).

This represents the sum of the approximate number of CPU seconds of program execution time while the transactions were active.
DLI database calls
The total number of DL/I database calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate total number of DL/I database calls that the program issued while the transactions were active.

Fast path DEDB calls
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.

Fast path MSDB calls
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.
IMS Transaction Utilization Detail by Date report

For the day selected, this report shows the resources used and the response times for the transactions selected. Figure 44 shows an example of a report.

This information identifies the report:

<table>
<thead>
<tr>
<th>Report ID</th>
<th>IMST01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report group</td>
<td>IMS reports</td>
</tr>
<tr>
<td>Source</td>
<td>IMS_TRANSACTION_D</td>
</tr>
<tr>
<td>Attributes</td>
<td>IMS, transaction, utilization, performance, daily</td>
</tr>
<tr>
<td>Variables</td>
<td>MVS_SYSTEM_ID, IMS_SYSTEM_ID, IMS_TRANSACTION_NAME, DATE</td>
</tr>
</tbody>
</table>

IMS Transaction Utilization Detail by Date
Date: 2000-02-26
System: 'MVS2' IMS System: 'IMS2'

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>EMH response time</th>
<th>MSGQ response time</th>
<th>Average EMH response time</th>
<th>Average MSGQ response time</th>
<th>Program CPU time</th>
<th>DLI database calls</th>
<th>Fast Path DEDB calls</th>
<th>Fast Path MSDB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSSWCH</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MSGSWCH</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CIT201</td>
<td>0</td>
<td>161</td>
<td>0.00</td>
<td>0.45</td>
<td>10.878</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC201</td>
<td>0</td>
<td>207</td>
<td>0.00</td>
<td>0.93</td>
<td>11.070</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC5A01</td>
<td>0</td>
<td>190</td>
<td>0.00</td>
<td>1.19</td>
<td>9.797</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC5B01</td>
<td>0</td>
<td>190</td>
<td>0.00</td>
<td>1.19</td>
<td>9.797</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for OS/390 Report: IMST01

Figure 44. Example of IMS Transaction Utilization Detail by Date tabular report

The report contains this information:

Transaction name
The name of the transaction the user requested.

For descriptions of the other columns in this report, see “IMS User ID Response Time and CPU Detail by Date report” on page 181.
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 45 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

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 46 shows an example of a report.

This information identifies the report:

<table>
<thead>
<tr>
<th>Report ID</th>
<th>IMSS02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report group</td>
<td>IMS reports</td>
</tr>
<tr>
<td>Source</td>
<td>IMS_CHKPT_IOSAM_T</td>
</tr>
<tr>
<td>Attributes</td>
<td>IMS, buffers, utilization, performance, daily, OSAM, ISAM</td>
</tr>
<tr>
<td>Variables</td>
<td>MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME</td>
</tr>
</tbody>
</table>

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

Figure 46. 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.
Reports

**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 47 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

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 Region Utilization Detail by Date report

This report shows the resources used and response times delivered by each IMS dependent region, for the date selected. Figure 48 shows an example of a report.

This information identifies the report:

Report ID         IMSY05
Report group      IMS reports
Source            IMS_SYSTEM_D
Attributes        IMS, region, utilization, performance, daily
Variables         MVS_SYSTEM_ID, IMS_SYSTEM_ID, REGION_JOB_NAME, DATE

IMS Region Utilization Detail by Date
Date: 2000-02-26
System: 'MVS2' IMS System: 'IMS2'

Average Average Fast Fast
Region EMH MSGQ EMH MSGQ Program DLI Path Path
job response response response response CPU database DEDB MSDB
name time time time time time calls calls calls
-------- ---------- ---------- -------- -------- ---------- ---------- ---------- ----------
IMSSWCH 0 0 0.00 0.00 0.00 0000
MSGSWCH 0 0 0.00 0.00 0.00 0000
IMSR 0 1041 0.00 0.65 60.73 1000

Figure 48. Example of IMS Region Utilization Detail by Date tabular report

The report contains this information:

Region job name
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.

For descriptions of the other columns in this report, see "IMS User ID Response Time and CPU Detail by Date report" on page 181
IMS Region Detail by Date report

This report shows the resources used and response times delivered by each IMS dependent region for each transaction class that ran in that region, for the date selected. Figure 49 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY06
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_D
- **Attributes**: IMS, region, transaction, utilization, performance, daily
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, REGION_JOB_NAME, TRANSACTION_CLASS, DATE

IMS Region Detail by Date
Date: 2000-02-26
System: 'MVS2' IMS System: 'IMS2'

<table>
<thead>
<tr>
<th>Region job name</th>
<th>Tr cl</th>
<th>EMH</th>
<th>MSGQ</th>
<th>EMH</th>
<th>MSGQ</th>
<th>Average EMH response time</th>
<th>Average MSGQ response time</th>
<th>Program CPU calls</th>
<th>DLI Path calls</th>
<th>Fast Path DEDB calls</th>
<th>Fast Path MSDB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMSSWCH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSGSWCH</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>1</td>
<td>0</td>
<td>161</td>
<td>0.00</td>
<td>0.45</td>
<td>10.87</td>
<td>8000</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>3</td>
<td>0</td>
<td>175</td>
<td>0.00</td>
<td>1.08</td>
<td>9.875</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>5</td>
<td>0</td>
<td>107</td>
<td>0.00</td>
<td>0.28</td>
<td>6.827</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>2</td>
<td>0</td>
<td>154</td>
<td>0.00</td>
<td>0.96</td>
<td>10.497</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>4</td>
<td>0</td>
<td>207</td>
<td>0.00</td>
<td>0.93</td>
<td>11.070</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>1</td>
<td>0</td>
<td>46</td>
<td>0.00</td>
<td>0.29</td>
<td>1.786</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMSR</td>
<td>2</td>
<td>0</td>
<td>190</td>
<td>0.00</td>
<td>1.19</td>
<td>9.797</td>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 49. Example of IMS Region Detail by Date tabular report

The report contains this information:

**Region job name**

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.

**Tr cl**

The assigned transaction class.

For descriptions of the other columns in this report, see "IMS User ID Response Time and CPU Detail by Date report" on page 181.
**IMS Resource Detail by Quarter Hour report**

This report shows the resources used and response times for the selected IMS system at 15 minute intervals during a specified time period. Figure 50 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSY07
- **Report group**: IMS reports
- **Source**: IMS_SYSTEM_Q
- **Attributes**: IMS, utilization, performance, volumes, response
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME

The report contains this information:

**Time**
The start time of a 15-minute interval for which the data applies.

For descriptions of the other columns in this report, see “IMS User ID Response Time and CPU Detail by Date report” on page 181.

<table>
<thead>
<tr>
<th>Time</th>
<th>EMM response time</th>
<th>MSGQ response time</th>
<th>Average EMM response time</th>
<th>Average MSGQ response time</th>
<th>Program CPU time</th>
<th>DLI database calls</th>
<th>Fast Path calls</th>
<th>Fast Path calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>16.15.00</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16.30.00</td>
<td>0</td>
<td>973</td>
<td>0.00</td>
<td>0.65</td>
<td>56.707</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16.45.00</td>
<td>0</td>
<td>68</td>
<td>0.00</td>
<td>0.64</td>
<td>4.023</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 50. Example of IMS Resource Detail by Quarter Hour tabular report

The report contains this information:

**Time**
The start time of a 15-minute interval for which the data applies.
IMS User ID Resource Worst Case by Date report

This report shows the resources used by the most resource intensive transactions run by the selected user ID, for the date selected. Figure 51 shows an example of a report.

This information identifies the report:

Report ID  IMST02
Report group  IMS reports
Source  IMS_USER_TRAN_D
Attributes  IMS, user ID, transaction, utilization, performance, daily, worst
Variables  MVS_SYSTEM_ID, IMS_SYSTEM_ID, USER_ID, TRANSACTION_NAME, DATE

IMS User ID Resource Worst Case by Date
Date: 2000-02-26
System: 'MVS2' IMS System: 'IMS2'
User ID: 'DSWP0001'

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>EMH response time</th>
<th>MSGQ response time</th>
<th>Average EMH response time</th>
<th>Average MSGQ response time</th>
<th>Program CPU time</th>
<th>Database calls</th>
<th>DEOB calls</th>
<th>MSOB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSE0101</td>
<td>0</td>
<td>26</td>
<td>0.00</td>
<td>1.14</td>
<td>1.41</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC201</td>
<td>0</td>
<td>16</td>
<td>0.00</td>
<td>0.96</td>
<td>1.11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CIT201</td>
<td>0</td>
<td>12</td>
<td>0.00</td>
<td>0.39</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC5A01</td>
<td>0</td>
<td>18</td>
<td>0.00</td>
<td>1.07</td>
<td>0.85</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CIT801</td>
<td>0</td>
<td>16</td>
<td>0.00</td>
<td>1.26</td>
<td>0.78</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC101</td>
<td>0</td>
<td>9</td>
<td>0.00</td>
<td>0.25</td>
<td>0.68</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC5B01</td>
<td>0</td>
<td>4</td>
<td>0.00</td>
<td>0.22</td>
<td>0.19</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 51. Example of IMS User ID Resource Worst Case by Date tabular report

The report contains this information:

**Transaction name**

The name of the IMS transaction the user requested. For full function activity, this is the name of the scheduler message block (SMB). For Fast Path activity, this column is the routing code. For APPC activity, this column is the transaction program instance (TPI).

For descriptions of the other columns in this report, see “IMS User ID Response Time and CPU Detail by Date report” on page 181.
IMS Program Utilization Worst Case by Date report

For the day selected, this report shows the resources used by the programs. Figure 52 shows an example of a report.

This information identifies the report:

**Report ID** IMST03

**Report group** IMS reports

**Source** IMS_TRANSACTION_D

**Attributes** IMS, program, utilization, performance, daily, worst

**Variables** MVS_SYSTEM_ID, IMS_SYSTEM_ID, PROGRAM_NAME, DATE

<table>
<thead>
<tr>
<th>Program name</th>
<th>EMM response time</th>
<th>MSGQ response time</th>
<th>Average EMM response time</th>
<th>Average MSGQ response time</th>
<th>Program CPU time</th>
<th>DLI database calls</th>
<th>Fast Path DDB calls</th>
<th>Fast Path MSDB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>PPSC5A01</td>
<td>0</td>
<td>207</td>
<td>0.00</td>
<td>0.93</td>
<td>11.070</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPIT2A01</td>
<td>0</td>
<td>161</td>
<td>0.00</td>
<td>0.45</td>
<td>10.878</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPSC2A01</td>
<td>0</td>
<td>154</td>
<td>0.00</td>
<td>0.96</td>
<td>10.497</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPIT8A01</td>
<td>0</td>
<td>175</td>
<td>0.00</td>
<td>1.08</td>
<td>9.875</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPS1A01</td>
<td>0</td>
<td>190</td>
<td>0.00</td>
<td>1.19</td>
<td>9.797</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPSC1A01</td>
<td>0</td>
<td>107</td>
<td>0.00</td>
<td>0.28</td>
<td>6.827</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PPSC5B01</td>
<td>0</td>
<td>46</td>
<td>0.00</td>
<td>0.29</td>
<td>1.786</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>IMSWCH</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MSGWCH</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td>0.00</td>
<td>0.000</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 52. Example of IMS Program Utilization Worst Case by Date tabular report

The report contains this information:

**Program name**

The name of the IMS application program used to process the transaction. For full function and Fast Path activity, this column is the program specification block (PSB) if available. For APPC activity this column is the TPI used.

For descriptions of the other columns in this report, see “IMS User ID Response Time and CPU Detail by Date report” on page 181.
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

The report contains this information:

Date: '2002-09-27 to '2002-09-30'
IMS System: 'IMS 71A'

<table>
<thead>
<tr>
<th>Date</th>
<th>Up Schedule Hours</th>
<th>Up In Schedule %</th>
<th>Objective %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-09-27</td>
<td>12</td>
<td>9</td>
<td>100.00</td>
</tr>
<tr>
<td>2002-09-28</td>
<td>24</td>
<td>9</td>
<td>100.00</td>
</tr>
<tr>
<td>2002-09-29</td>
<td>24</td>
<td>9</td>
<td>100.00</td>
</tr>
<tr>
<td>2002-09-30</td>
<td>3</td>
<td>3</td>
<td>33.33</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for OS/390 Report: CSQA01

Figure 53. 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

<table>
<thead>
<tr>
<th>Region Name</th>
<th>Up Hours</th>
<th>Schedule Hours</th>
<th>Up In Schedule</th>
<th>Up In Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION1</td>
<td>12</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
<tr>
<td>REGION2</td>
<td>12</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
</tbody>
</table>

**Figure 54. 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'

<table>
<thead>
<tr>
<th>Application Name</th>
<th>Up Hours</th>
<th>CPU Usage Hours</th>
<th>Up In Schedule Hours</th>
<th>Up In Schedule %</th>
<th>Objective %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applic_1</td>
<td>12</td>
<td>1</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
<tr>
<td>Applic_2</td>
<td>12</td>
<td>3</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for OS/390 Report: CSQA03

---

Figure 55. 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.
<table>
<thead>
<tr>
<th>Objective (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability objective for the IMS subsystem related to this application in the scheduled hours.</td>
</tr>
</tbody>
</table>
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 56 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

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.

**Figure 56. 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.
### Reports

<table>
<thead>
<tr>
<th>ShMsg</th>
<th>Average Full Function Short messages on queue before processing.</th>
</tr>
</thead>
<tbody>
<tr>
<td>LgMsg</td>
<td>Average Full Function Long messages on queue before processing.</td>
</tr>
<tr>
<td>Processed</td>
<td>Total number of messages processed.</td>
</tr>
<tr>
<td>AvgTime</td>
<td>Average time (in seconds) each message remains on queue.</td>
</tr>
</tbody>
</table>
IMS Message 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 57 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

The report contains this information:

| Date | 2002-12-05 |
| Time | 15:15, 15:30 |
| IMS System | 'CSSD' |
| Queue Type | 'MSGQ LOCAL' |
| Transaction Name | HC9CHKRT, HC9PMUT0, HC9SWFT0, HC9T100T, HC9T200T, HC9CRIT0, HC9T110T, HEQAAUU2, HEQBW104 |
| Time | 15:15, 15:30 |
| Message Queues Input | FPMsg, SHMsg, LGMsg |
| Message Queues Output | Processed, AvgTime |
| Processed | AvgTime |

Figure 57. 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 201
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 58 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

<table>
<thead>
<tr>
<th>Date</th>
<th>Queue Type</th>
<th>FpMsg</th>
<th>ShMsg</th>
<th>LgMsg</th>
<th>Processed</th>
<th>AvgTime</th>
<th>FpMsg</th>
<th>ShMsg</th>
<th>LgMsg</th>
<th>Processed</th>
<th>AvgTime</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-07-17</td>
<td>EMHQ LOCAL</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0.00</td>
<td>1.00</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0.63</td>
</tr>
<tr>
<td>2002-07-17</td>
<td>MSGQ LOCAL</td>
<td>-</td>
<td>0.24</td>
<td>10.56</td>
<td>3841</td>
<td>0.36</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3609</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Figure 58. 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 201.
IMS Transaction Arrival Rate and Message 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 59 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

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 201.
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 60 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

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>Total Transactions</th>
<th>CPU Average</th>
<th>Input</th>
<th>Input</th>
<th>Input</th>
<th>Input</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Approximate</td>
<td>Response</td>
<td>Local</td>
<td>CQS</td>
<td>Queue</td>
</tr>
<tr>
<td></td>
<td></td>
<td>time</td>
<td>time</td>
<td>time</td>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>A70200</td>
<td>6.8300E+02</td>
<td>3.903E+00</td>
<td>1.878E-01</td>
<td>6.830E+02</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>A70210</td>
<td>2.0000E+00</td>
<td>0.000E+00</td>
<td>1.500E-01</td>
<td>2.000E+00</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>A7041030</td>
<td>4.2000E+01</td>
<td>4.748E-03</td>
<td>3.333E-02</td>
<td>4.200E+01</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>A7042030</td>
<td>4.6000E+01</td>
<td>4.845E-03</td>
<td>3.804E-01</td>
<td>4.600E+01</td>
<td>0.000E+00</td>
<td></td>
</tr>
<tr>
<td>A7051030</td>
<td>1.3600E+02</td>
<td>1.299E-02</td>
<td>1.390E-01</td>
<td>1.360E+02</td>
<td>0.000E+00</td>
<td></td>
</tr>
</tbody>
</table>

**Average**

<table>
<thead>
<tr>
<th>Average</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>Output</td>
</tr>
<tr>
<td>Process</td>
<td>Local</td>
</tr>
<tr>
<td>time</td>
<td>time</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>1.025E-03</td>
<td>1.114E-01</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>3.333E-02</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>2.913E-01</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>6.912E-02</td>
</tr>
</tbody>
</table>

**Average Total Network Responses**

<table>
<thead>
<tr>
<th>Average Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.630E+02</td>
</tr>
<tr>
<td>2.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
</tr>
<tr>
<td>4.600E+01</td>
</tr>
<tr>
<td>1.340E+02</td>
</tr>
</tbody>
</table>

**Figure 60. 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-Shared Queue 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 Shared Queue.

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

Output CQS Queue  The total number of output messages issued by transactions and BMP programs queued through IMS Shared Queue. Always zero in a non-Shared Queue 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-Shared Queue 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 OS/390 with QMF (QMFUSE=YES coded in the Tivoli Decision Support for OS/390 initialization member DRLFPROF). Figure 61 on page 210 shows an example of a report.

This information identifies the report:

<table>
<thead>
<tr>
<th>Report ID</th>
<th>CSQTQ06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report group</td>
<td>CSQ reports</td>
</tr>
<tr>
<td>Source</td>
<td>IMS_TRAN_D</td>
</tr>
<tr>
<td>Attributes</td>
<td>IMS, Lterm, Utilization, Performance, Daily</td>
</tr>
<tr>
<td>Variables</td>
<td>Origin_IMS, Process_IMS, Date, origin_Lterm, User_ID</td>
</tr>
</tbody>
</table>
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-Shared Queue 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 Shared Queue.

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

Output CQS Queue
The total number of output messages issued by transactions and BMP programs queued through IMS Shared Queue. Always zero in a non-Shared Queue 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 Shared Queue 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 OS/390 with QMF (QMFUSE=YES coded in the Tivoli Decision Support for OS/390 initialization member DRLFPROF). Figure 62 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

The report contains this information:

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

<table>
<thead>
<tr>
<th>Region</th>
<th>Job Program</th>
<th>Total Utilization</th>
<th>Average Input</th>
<th>Input Time</th>
<th>Input Queue</th>
<th>Input CQS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPPEROB1</td>
<td>DO31</td>
<td>3.1000E+01</td>
<td>0.000E+00</td>
<td>2.484E-01</td>
<td>3.100E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO35</td>
<td>1.0000E+01</td>
<td>2.845E-03</td>
<td>2.070E+00</td>
<td>1.000E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO40</td>
<td>3.1000E+01</td>
<td>7.964E-03</td>
<td>1.097E+01</td>
<td>3.100E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO41</td>
<td>9.0000E+00</td>
<td>2.658E-03</td>
<td>2.444E-01</td>
<td>9.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO675</td>
<td>1.8000E+01</td>
<td>3.441E-03</td>
<td>6.111E-01</td>
<td>1.800E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO676</td>
<td>3.5000E+01</td>
<td>8.235E-03</td>
<td>6.000E-01</td>
<td>3.500E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td></td>
<td>DO700</td>
<td>1.6000E+01</td>
<td>3.127E-03</td>
<td>5.062E-01</td>
<td>1.600E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>FSRASF1I</td>
<td>1.9700E+02</td>
<td>4.350E-02</td>
<td>1.229E+00</td>
<td>1.970E+02</td>
<td>0.000E+00</td>
<td></td>
</tr>
</tbody>
</table>

Average:

<table>
<thead>
<tr>
<th>Average Input</th>
<th>Average Output</th>
<th>Average Input</th>
<th>Average Output</th>
<th>Average Input</th>
<th>Average Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.484E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>5.556E-02</td>
<td>5.556E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>1.333E-01</td>
<td>1.111E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>6.138E-02</td>
<td>5.573E-01</td>
<td>1.630E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

Average:

<table>
<thead>
<tr>
<th>Average Output</th>
<th>Average Input</th>
<th>Average Output</th>
<th>Average Input</th>
<th>Average Output</th>
<th>Average Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
</tbody>
</table>

Figure 62. Example of IMS CSQ Transit Time Analysis By Region, Daily report.
ORIGIN IMS
The IMS subsystem ID defined in the origin part of the UOW token. It identifies the activity origin. In a non-Shared Queue 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 Shared Queue.

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

Output CQS Queue
The total number of output messages issued by transactions and BMP programs queued through IMS Shared Queue. Always zero in a non-Shared Queue 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-Shared Queue 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 63 on page 217 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

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>Total Transactions</th>
<th>Program CPU time</th>
<th>DC calls GN</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BMP</td>
<td>0.00000E+00</td>
<td>2.00521E-01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>AUSWAHL</td>
<td>4.44000E+02</td>
<td>1.02109E+00</td>
<td>4.440E+02</td>
</tr>
<tr>
<td>A5760010</td>
<td>7.20000E+01</td>
<td>1.51792E+00</td>
<td>1.440E+02</td>
</tr>
<tr>
<td>A5770010</td>
<td>3.00000E+00</td>
<td>6.69531E-02</td>
<td>6.000E+00</td>
</tr>
<tr>
<td>A5780010</td>
<td>1.00000E+00</td>
<td>2.27839E-01</td>
<td>3.000E+00</td>
</tr>
<tr>
<td>A5790010</td>
<td>1.00000E+00</td>
<td>5.07031E-02</td>
<td>2.000E+00</td>
</tr>
<tr>
<td>A5920010</td>
<td>1.68000E+02</td>
<td>3.31427E+00</td>
<td>3.320E+02</td>
</tr>
<tr>
<td>A702000</td>
<td>6.83000E+02</td>
<td>3.90340E+00</td>
<td>6.830E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB calls</th>
<th>DB calls</th>
<th>DB calls</th>
<th>DB calls</th>
<th>DB calls</th>
<th>DB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLET</td>
<td>GHN</td>
<td>GHNP</td>
<td>GHU</td>
<td>GN</td>
<td>GNP</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>2.161E+03</td>
<td>0.000E+00</td>
<td>1.500E+01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>4.440E+02</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>5.430E+02</td>
<td>5.430E+02</td>
<td>0.000E+00</td>
<td>5.940E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DB calls</th>
<th>DB calls</th>
<th>DB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>GU</td>
<td>ISRT</td>
<td>REPL</td>
</tr>
<tr>
<td>9.000E+00</td>
<td>1.500E+01</td>
<td>1.800E+01</td>
</tr>
<tr>
<td>4.440E+02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>6.830E+02</td>
<td>0.000E+00</td>
<td>1.086E+03</td>
</tr>
</tbody>
</table>

---

**Figure 63. Example of IMS CSQ Resource Utilization, Daily Overview**

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**  
The 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.
Reports

- **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.
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 64 on page 220 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

Figure 64. Sample JCL for assembling IMS log record DSECTs (Part 1 of 3)
Figure 64. Sample JCL for assembling IMS log record DSECTs (Part 2 of 3)
Figure 64. Sample JCL for assembling IMS log record DSECTs (Part 3 of 3)
Appendix C. DRLJXIDC DSECT macro

This appendix shows the layout of the output composite record produced by DRL2LOGP.

Figure 65 on page 224 shows the assembler DSECT macro DRLJXIDC. You can use this macro, in conjunction with the IMS ILOGREC DSECTs, if you are writing your own record procedure to read the composite records produced by the log procedure. It contains the layout of the output composite record written to ddname DRLICOMP by DRL2LOGP.

The source for this macro is in member DRLJXIDC in the Tivoli Decision Support for OS/390 CNTL library.

For information about writing Tivoli Decision Support for OS/390 record procedures, refer to the Language Guide and Reference.
DRLJXIDC DSECT macro

MACRO
DRLJXIDC &PFX=BASIC,&TYPE=DSECT
AIF ('&TYPE' EQ 'DSECT' OR T'&TYPE EQ 'O').NEXT1
&RTYPE SETC 'EQU'
&O SETC 'x'
AGO .BR
NEXT1 ANOP
&RTYPE SETC 'DSECT'
&O SETC ' ' .BR
ANOP
AIF ('&PFX' EQ 'BASIC').LBASIC
AIF ('&PFX' EQ 'D1').LD1
AIF ('&PFX' EQ 'D2').LD2
AIF ('&PFX' EQ 'PSB').LPSB
AIF ('&PFX' EQ 'SPA').LSPA
AIF ('&PFX' EQ 'FFSP').LFFSP
AIF ('&PFX' EQ 'FPSP').LFPSP
AIF ('&PFX' EQ 'ESS').LESS
AIF ('&PFX' EQ 'EMH').LEMH
MNOTE 8,'INVALID PREFIX TYPE SPECIFIED'
MEXIT
LBASIC ANOP
CMPREC &RTYPE &O
**************************************************************************
* BASIC PREFIX OF COMPOSITE RECORDS
**************************************************************************
CMLL DS H LENGTH OF LOG RECORD
CMZZ DS H HALF WORD OF ZERO
CMTYPE DS C COMPOSITE REC TYPE
CMSBTYPE DS C COMPOSITE REC SUB TYPE
CMFLAG1 DS C
CMFWI EQU X'80' TXN IS WFI
CMFAOI EQU X'40' TXN STARTED BY AOI EXIT
CMFPNT EQU X'20' PARENT TRANSACTION
CMFROOT EQU X'10' ROOT TRANSACTION
CMFPSEND EQU X'08' PSEUDO SCHEDULE
CMFEND EQU X'04' PSEUDO END
CMFABND EQU X'02' TRANSACTION ABENDED
CMFIRENQ EQU X'01' TRANSACTION RE-ENQUEUED
CMFLOG2 DS C
CMFICNV EQU X'80' START OF CONVERSATION
* EQU X'40' RESERVED FOR FUTURE USE
* EQU X'20' RESERVED FOR FUTURE USE
* EQU X'10' RESERVED FOR FUTURE USE
* EQU X'08' RESERVED FOR FUTURE USE
* EQU X'04' RESERVED FOR FUTURE USE
* EQU X'02' RESERVED FOR FUTURE USE
* EQU X'01' RESERVED FOR FUTURE USE
CMIMSID DS CL8 IMS ID
CMFDATE DS PL4 DATE STAMP OF PSEUDO END
CMFTIME DS PL4 TIME STAMP OF PSEUDO END
CMFQ6TM DS F SQ6 TIME OF PSEUDO END
CMSMB DS OCL12 SMB DATA
CMSMBNM DS CL8 SMB NAME
CMSMBCL DS CL1 SMB CLASS
CMSMBPR DS CL1 SMB PRIORITY
CMSMBPL DS H SMB PROCESSING LIMIT
CMROOT DS OCL8 ROOT INPUT SECTION TRIPLET

Figure 65. DRLJXIDC MACRO (Part 1 of 5)
CMROFF DS F OFFSET OF ROOT INPUT SECTION
CMRTL DS H LENGTH OF ROOT INPUT SECTION
CMRNO DS H NUMBER OF ROOT INPUT SECTION
CMIINPUT DS OCL8 INPUT SECTION TRIPLET
CMIOFFSET DS F OFFSET OF INPUT SECTION
CMINL DS H LENGTH OF INPUT SECTION
CMINNO DS H NUMBER OF INPUT SECTION
CMSPAPUT DS OCL8 SPA SECTION TRIPLET
CMSPAOFF DS F OFFSET OF SPA SECTION
CMSPAL DS H LENGTH OF SPA SECTION
CMSPANO DS H NUMBER OF SPA SECTION
CMIOUTPUT DS OCL8 OUTPUT SECTION TRIPLET
CMIOUTOFF DS F OFFSET OF OUTPUT SECTION
CMIOUTL DS H LENGTH OF OUTPUT SECTION
CMIOUTNO DS H NUMBER OF OUTPUT SECTION
CMSPBPUT DS OCL8 PSB SECTION TRIPLET
CMSPBOFF DS F OFFSET OF PSB SECTION
CMSPBL DS H LENGTH OF PSB SECTION
CMSPBNO DS H NUMBER OF PSB SECTION
CMFFSP DS OCL8 FF SYNCPOINT SECTION TRIPLET
CMFFOFF DS F OFFSET OF FF-SP SECTION
CMFFL DS H LENGTH OF FF-SP SECTION
CMFFNO DS H NUMBER OF FF-SP SECTION
CMFPSP DS OCL8 FP SYNCPOINT SECTION TRIPLET
CMFPOFF DS F OFFSET OF FP-SP SECTION
CMFPL DS H LENGTH OF FP-SP SECTION
CMFPNO DS H NUMBER OF FP-SP SECTION
CMESS DS OCL8 EXT SUBSYSTEM SECTION TRIPLET
CMESOFF DS F OFFSET OF ESS SECTION
CMESL DS H LENGTH OF ESS SECTION
CMESSNO DS H NUMBER OF ESS SECTION
CMEMH DS OCL8 EMH SECTION TRIPLET
CMEMHOFF DS F OFFSET OF EMH SECTION
CMEMHL DS H LENGTH OF EMH SECTION
CMEMHNO DS H NUMBER OF EMH SECTION
CMBPLEN EQU *-CMPREC LENGTH OF BASIC PREFIX
MEXIT
LDI ANOP
CMDSSECT &RTYPE &0
******************************************************************************
*                        PREFIX OF THE DI SECTION                           *
******************************************************************************
CMD1ID DS CL2 EYE CATCHER FOR DI SECTION ('D1')
CMD1PRL LENGTH OF DI SECTION PREFIX
CMD1L DS H LENGTH OF DI SECTION
CMD1FLG1 DS C
CMD1X01 EQU X'80' X01/X03 RECORD(S) PRESENT
CMD1X35 EQU X'40' X35 RECORD(S) PRESENT
CMD1X31 EQU X'20' X31 RECORD PRESENT
CMD1X36 EQU X'10' X36 RECORD(S) PRESENT
CMD1MCAN EQU X'08' MESSAGE CANCELLED (X34 PRESENT)
* EQU X'04' RESERVED FOR FUTURE USE
* EQU X'02' RESERVED FOR FUTURE USE
* EQU X'01' RESERVED FOR FUTURE USE

Figure 65. DRLJXIDC MACRO (Part 2 of 5)
DRLJXIDC DSECT macro

CMD1FLG2 DS C
  * EQU X'80' RESERVED FOR FUTURE USE
  * EQU X'40' RESERVED FOR FUTURE USE
  * EQU X'20' RESERVED FOR FUTURE USE
  * EQU X'10' RESERVED FOR FUTURE USE
  * EQU X'08' RESERVED FOR FUTURE USE
  * EQU X'04' RESERVED FOR FUTURE USE
  * EQU X'02' RESERVED FOR FUTURE USE
  * EQU X'01' RESERVED FOR FUTURE USE
CMD1NOSG DS H NUMBER OF INPUT SEGMENTS
CMD1OUTS DS H NUMBER OF ACTUAL OUTPUTS
MEXIT
LD2 ANOP
CMD2SECT &RTYPE &D
******************************************************************************
* PREFIX OF THE D2 SECTION
*
******************************************************************************
CMD2ID DS CL2 EYE CATCHER FOR D2 SECTION ('D2')
CMD2PRLL DS H LENGTH OF D2 SECTION PREFIX
CMD2LL DS H LENGTH OF D2 SECTION
CMD2FLG1 DS C
CMD2X03 EQU X'80' X03 RECORD(S) PRESENT
CMD2X35 EQU X'40' X35 RECORD(S) PRESENT
CMD2X31 EQU X'20' X31 RECORD PRESENT
CMD2X36 EQU X'10' X36 RECORD(S) PRESENT
CMD2PGSW EQU X'08' PROGRAM SWITCH (DEST IS SMB)
CMD2MCAN EQU X'04' MESSAGE CANCELLED (X34 PRESENT)
  * EQU X'02' RESERVED FOR FUTURE USE
  * EQU X'01' RESERVED FOR FUTURE USE
CMD2FLG2 DS C
  * EQU X'80' RESERVED FOR FUTURE USE
  * EQU X'40' RESERVED FOR FUTURE USE
  * EQU X'20' RESERVED FOR FUTURE USE
  * EQU X'10' RESERVED FOR FUTURE USE
  * EQU X'08' RESERVED FOR FUTURE USE
  * EQU X'04' RESERVED FOR FUTURE USE
  * EQU X'02' RESERVED FOR FUTURE USE
  * EQU X'01' RESERVED FOR FUTURE USE
CMD2NOSG DS H NUMBER OF OUTPUT SEGMENTS
DS CL2 RESERVED FOR FUTURE USE
MEXIT
LSPA ANOP
CMSPASEC &RTYPE &D

Figure 65. DRLJXIDC MACRO (Part 3 of 5)
Figure 65. DRLJXIDC MACRO (Part 4 of 5)
DRLJXIDC DSECT macro

*****************************************************************
* PREFIX OF THE ESS SECTION
*****************************************************************
CMESSID DS CL2 EYE CATCHER FOR ESS SECTION ('ES')
CMESSPLL DS H LENGTH OF ESS SECTION PREFIX
CMESSLL DS H LENGTH OF ESS SECTION
CMESSFL1 DS C
* EQU X'80' RESERVED FOR FUTURE USE
* EQU X'40' RESERVED FOR FUTURE USE
* EQU X'20' RESERVED FOR FUTURE USE
* EQU X'10' RESERVED FOR FUTURE USE
* EQU X'08' RESERVED FOR FUTURE USE
* EQU X'04' RESERVED FOR FUTURE USE
* EQU X'02' RESERVED FOR FUTURE USE
* EQU X'01' RESERVED FOR FUTURE USE
CMESSFLG2 DS C
* EQU X'80' RESERVED FOR FUTURE USE
* EQU X'40' RESERVED FOR FUTURE USE
* EQU X'20' RESERVED FOR FUTURE USE
* EQU X'10' RESERVED FOR FUTURE USE
* EQU X'08' RESERVED FOR FUTURE USE
* EQU X'04' RESERVED FOR FUTURE USE
* EQU X'02' RESERVED FOR FUTURE USE
* EQU X'01' RESERVED FOR FUTURE USE
CMESSCNT DS H NUMBER OF ESS RECS FOUND
DS CL2 RESERVED FOR FUTURE USE
MEXIT
LEMH ANOP
CMEMHSEC &RTYPE &O

Figure 65. DRLJXIDC MACRO (Part 5 of 5)
Appendix D. Sample archive exit

This appendix shows a sample IMS archive exit, DRLJXIMS. The source for this sample exit can also be found in member DRLJXIMS in the Tivoli Decision Support for OS/390 CNTL library. Figure 66 on page 230 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 Tivoli Decision Support for *
* OS/390/IMS *
* COPYRIGHT: NONE *
* STATUS: IMS/ESA V3R1 *
* FUNCTION: *
* WRITES THE RECORDS USED BY Tivoli Decision Support for *
* OS/390/IMS TO THE FILE DEFINED BY THE DDNAME IMSLOG. *
* THE RECORD TYPES CONCERNED ARE AS FOLLOWS (ALL IN HEX) *
* 01, 02, 03, 07, 08, 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 Tivoli Decision *
* Support for OS/390/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, 07, 08, 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, 07, 08, 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. *

Figure 66. Sample IMS archive exit

230 Tivoli Decision Support for OS/390: IMS Performance Feature Guide and Reference
Appendix E. DFSLMG0 log merge utility

The DFSLMG0 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. DFSLMG0 is placed in IMS.RESLIB during IMS system definition.

Following is the description of the statements:

STEPLIB DD
It 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.LOBG,DISP=OLD,
// VOL=SER=XXXXXX, UNIT=TAPE

LOGOUT DD
Describes the output data set.
//LOGOUT DD DSNAME=IMS.LOGOUT,DISP=(NEW,CATLG)
// VOL=SER=yyyy, UNIT=3390,SPACE=(CYL, (100,5)),
// DCB==(RECFM=VB, LRECL=27994, BLKSIZE=32760)

SYSIN DD
Describes the control statement data set.
//SYSIN DD *
**DFSLTMG0 log merge utility**

**START**
Used to specify a start time. This statement must be present (yyddd, hhmmssstt).

**STOP**

**Log Record Selection**
Use this control statement to merge only certain types of log records.

---

**Controlling the log merge**

To control the log output, you need to:
- Choose the required systems to participate in the logical link paths you want to examine.
- Coordinate the series of input logs for each system, so that they cover a similar time span.
- 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.
- 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 F. List of abbreviations

These abbreviations appear in this book:

- API: application program interface
- APPC: advanced program-to-program communication
- APPN: Advanced peer to peer networking
- BMP: batch message processing program
- BSAM: basic sequential access method
- BTAM: basic telecommunications access method
- CCB: conversational control block
- CLB: communication line block
- CNT: communications name table
- CPI-C: common program interface for communications
- CPU: central processing unit
- CRB: communication restart block
- CSA: common storage area
- CTB: communication terminal block
- DASD: direct access storage device
- DB/DC: database/data communications
- DBCTL: database control facility
- DBD: database description
- DBRC: database recovery control
- DC: data communications
- DDIR: database directory
- DEDB: data entry database
- DL/I: Data Language 1
- DLISAS: DL/I separate address space
- DMB: database manager blocks
- DRRN: disk relative record number
- DSECT: dummy control section
- EMH: expedited message handler
- EMHB: EMH buffer
- ESS: External subsystem
- FES: front-end system
- FP: Fast Path
- GSAM: generalized sequential access method
GU  get unique
HDAM  hierarchical direct access method
HIDAM  hierarchical indexed direct access method
HISAM  hierarchical indexed sequential access method
HSAM  hierarchical sequential access method
IMSPARS  IMS Performance Analysis and Reporting System
I/O  input/output
IFP  IMS Fast Path
IMS/ESA  Information Management System/Extended System Architecture
IRLM  IMS resource locking
ISC  intersystem communication
JCL  job control language
LTERM  logical terminal
MPP  message processing program
MPR  message processing region
MSC  multiple systems coupling
MSDB  main storage database
MSGQ  message queue manager
MTO  master terminal operator
MVS  multiple virtual storage
OLDS  online log data set
OSAM  overflow sequential access method
OTMA  open transaction manager access
PCB  program control block
PDIR  PCB directory
PSB  program specification block
QPOOL  message queue buffers
QSAM  queued sequential access method
RACF  Resource Access Control Facility
RDS  recovery data set
RECANY  receive any buffer
RECON  recovery control data set
RLDS  recovery log data set
RTKN  recovery token
SHISAM  simple hierarchical indexed sequential access method
SHSAM  simple hierarchical sequential access method
SLDS  system log data set
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLR</td>
<td>Service Level Reporter</td>
</tr>
<tr>
<td>SMB</td>
<td>scheduler message block</td>
</tr>
<tr>
<td>SMF</td>
<td>System Management Facilities</td>
</tr>
<tr>
<td>SPA</td>
<td>scratch pad area</td>
</tr>
<tr>
<td>SQ</td>
<td>shared queue IMS</td>
</tr>
<tr>
<td>TPI</td>
<td>transaction program instance</td>
</tr>
<tr>
<td>TWS</td>
<td>Tivoli Workload Scheduler</td>
</tr>
<tr>
<td>UOR</td>
<td>unit of recovery</td>
</tr>
<tr>
<td>UOW</td>
<td>unit of work</td>
</tr>
<tr>
<td>VB</td>
<td>variable blocked</td>
</tr>
<tr>
<td>VSAM</td>
<td>virtual storage access method</td>
</tr>
<tr>
<td>VTAM</td>
<td>virtual telecommunications access method</td>
</tr>
<tr>
<td>WADS</td>
<td>write-ahead data set</td>
</tr>
<tr>
<td>WFI</td>
<td>wait for input</td>
</tr>
</tbody>
</table>
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
500 Columbus Avenue
Thornwood, NY 10594
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/or 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.

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.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information in softcopy form, the photographs and color illustrations might not appear.
Trademarks

DB2, DB2 Universal Database, IBM, the IBM logo, Tivoli, Tivoli Enterprise, Tivoli Enterprise Console, CICS, GDDM, IBMLink, IMS, IMS/ESA, MVS, Netview, OS/390, QMF, RACF, RMF, VTAM, WebSphere and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.
Glossary

A

administration. A Tivoli Decision Support for OS/390 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 OS/390.

C

collect. A process used by Tivoli Decision Support for OS/390 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 OS/390 database.

component. An optionally installable part of a Tivoli Decision Support for OS/390 feature.

control table. A predefined Tivoli Decision Support for OS/390 table that controls results returned by some log collector functions.

D

data table. A Tivoli Decision Support for OS/390 table that contains performance data used to create reports.

E

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.

I

internal data type. A data type used within Tivoli Decision Support for OS/390 during the processing of data.

K

key columns. The columns of a DB2 table that together constitute the key.

key value. Value that is used to sort records into groups.

L

log collector. A Tivoli Decision Support For OS/390 program that processes log data sets and provides other Tivoli Decision Support For OS/390 services.

log collector language. Tivoli Decision Support For OS/390 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 OS/390.

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 OS/390 DB2 table that contains grouping, translation, or substitution information.
**purge condition.** Instruction for purging old data from the Tivoli Decision Support For OS/390 database.

**record definition.** The description of a record type contained in the log data sets used by Tivoli Decision Support For OS/390, 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 OS/390 statements used to define reports and report groups.

**report group.** A collection of Tivoli Decision Support For OS/390 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 OS/390 DB2 table.

**system table.** A DB2 table that stores information that controls log collector processing, Tivoli Decision Support For OS/390 dialogs, and reporting.

**target.** In an update definition, the DB2 table in which Tivoli Decision Support For OS/390 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 OS/390 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

A
abend codes from transaction 43
abend during collect commit 43
accessibility xiv
active region statistics record definition 56
additional available IMS record information 43
administering the IMS Performance feature 31
AOI user exit initiated transaction special log procedure case 16, 111
APPC_MODE_NAME data column 71
APPC_NAME data column 71
APPC_NETID data column 70, 71
APPC_TIBQAB_A data column 71
APPC_TIBQAB_T data column 71
APPC_TPI_COMMITS data column 70
APPC_TPI_CPU_SEC data column 70
APPC_TPI_DPSBCALLS data column 70
APPC_TPI_ENDED data column 70
APPC_TPI_STARTED data column 70
application reports
  CPU utilization trend graphical 174
  response time overview
    graphical 167, 195, 197, 198, 201, 203, 204, 205, 206, 209, 213
    response time trend graphical 171
  transaction overview graphical 169
  transaction trend graphical 173
Application subcomponent tables 85
APPLICATION_NAME data column 98
APPLICATION_NAME key column 68, 85, 86
archive exit for IMS 229
attributes of reports 160
audience, intended xi

B
BACKGRND_WRITE_REQ data column 96
BLOCKS_WRITE_PURGE data column 88
BMP_DS_LMSG_BYTES data column 71
BMP_DS_LMSG_ISRRT data column 71
BMP_DS_SMSG_BYTES data column 71
BMP_DS_SMSG_ISRRT data column 71
BMP_FF_ABORTS data column 71
BMP_FF_COMMITS data column 71
BMP_FF_ABORTS data column 71
BMP_FF_COMMITS data column 71
BMP_OUTPUT_MSGS data column 71
BMP_PROCESS_SEC data column 71
BMP_PROGRAMS data column 71
books
  feedback xii
  online xii
  ordering xii
buffer pool reports
  detail for OSAM/ISAM by date 186
  detail for VSAM by date 188
BUFFER_ALTER_CALLS data column 88
BUFFER_ERRORS data column 96
BUFFER_ERRORS_MAX data column 96
BUFFER_SIZE data column 89
BUFFER_SIZE key column 87, 96
BUFFERS data column 88, 96
BUFFERS_CURRENT data column 89
BUFFERS_HIGH data column 89
BUFFERS_LOCKED data column 88
BUFFERS_OVERSIZE data column 89
BUFFERS_SEARCHED data column 88
CHKPT_LMSG_RECORDS data column 93
CHKPT_QBLK_RECORDS data column 93
CHKPT_SMSG_RECORDS data column 93
CI_FOUND_IN_POOL data column 96
CICSNAME parameter 31, 33
collect commit abend 43
collect for large installation 42
command log record definition 47
common reference columns 65
communications trace record definition 58
collection code from transaction 43
component
  installation 7, 107
  putting into production 9, 108
  selection 7, 107
composite record
  incomplete transaction 14
  record definition 59
  record sections 59
  record types and subtypes 64
  report from log procedure 19, 114
control tables
  DAY_OF_WEEK 66
  PERIOD_PLAN 66
conversation set record group 12
conversational transaction special log procedure case 16, 112
CPU and userid response time by date detail report 181
CPU utilization report
  application trend graphical 174
  system trend graphical 179
customer support xv
customizing the IMS Performance feature 6, 106
D
D summarization level suffix 65
D1 composite record section 60
D2 composite record section 60
data flow 23
  through DRL2LOGP 27, 117
  through log collector 25, 116
  through the IMS Performance feature 28
  through Tivoli Decision Support for OS/390 23
data tables
  built by IMS Shared Queue feature 135
  descriptions 65
  for application subcomponent 66
  for application subcomponent tables 85
  for IMS collect component 66
  for IMS log records component 66
data tables (continued)
for statistics subcomponent 66, 87
for system subcomponent 66, 83
for transaction subcomponent 66, 69
for transaction transit time component 135
IMS_APPLICATION_H, _W 85
IMS_CHKPT_JOSAM_T 87
IMS_CHKPT_POOLS_T 89
IMS_CHKPT_REGION_T 90
IMS_CHKPT_STATS_T 92
IMS_CHKPT_VSAM_T 96
IMS_PSB_ACCOUNT_H, _D, _W 148
IMS_SYSTEM_Q, _D 83
IMS_SYSTEM_tran_H, _D 143
IMS_TRAN_H, _D, _W 136
IMS_TRAN_QUEUE_Q, _D 140
IMS_TRANS_H, _D, _W 69
IMS_USER_APPL_D, _W 86
IMS_USER_TRAN_H, _D, _W 80
name suffix 65
naming standard 65
summarization level 65
data tables built by IMS Performance feature 65
database record definitions
buffer pool statistics 55
close 50
error 50
open 50
DATE key column 66, 68, 69, 80, 83, 85, 86, 87, 89, 90, 96
DAY_OF_WEEK control table 66
DB2 tables
built by IMS Shared Queue feature 135
DB2 tables built by IMS Performance feature 65
ddnames for DRL2LOGP
DRLDUMP 40, 128
DRLCHKI 40, 127
DRLCHKO 40, 127
DRLICOMP 40, 127
DRLIPARM 41, 128
DRLIRPTn 40, 128
DRLOUT 40, 127

defining the operating environment 5, 105
descriptions in tables 65
detail report 160
DFSCBT00 storage statistics record definition 56
discrete IMS sessions 42
dispatch statistics record definition 55
DLI utilization system trend graphical report 180
DLI_CMD_CALLS data column 71
DLI_EXCL_DEQUEUES data column 73
DLI_EXCL_ENQUEUE data column 73
DLI_EXCL_ENQWAITS data column 73
DLI_GCMD_CALLS data column 74
DLI_QCMD_CALLS data column 74
DLI_QCMD_ENQUEUE data column 74
DLI_QCMD_ENQWAITS data column 74
DLI_TEST_DEQUEUES data column 74
DLI_TEST_ENQUEUE data column 74
DLI_TEST_ENQWAITS data column 74
DLI_UPDT_DEQUEUES data column 74
DLI_UPDT_ENQUEUE data column 74
DLI_UPDT_ENQWAITS data column 74
DLIDB_CALLS data column 71
DLIDB_DELT_CALLS data column 72
DLIDB_GHN_CALLS data column 72
DLIDB_GHN_CALLS data column 72
DLIDB_DGU_CALLS data column 72
DLIDB_DGU_CALLS data column 72
DLIDB_ISRITER_CALLS data column 72
DLIDB_REPI_CALLS data column 73
DLIDBCTDBIO_SEC data column 72
DLIDBCTDBIOS data column 71
DLIDBCTLOCK_SEC data column 72
DLIDCGN_CALLS data column 73
DLIDC_GU_CALLS data column 73
DLIDC_ISRITER_CALLS data column 73
DLIDC_PURGE_CALLS data column 73
DLISq6_ACCUM_SEC data column 74
DMHR on I/O error record definition 58
DRL2LOGP
data flow 27, 117
data format 40, 127
ddnames 40, 127
description 40, 127
JCL example 41, 128
output record layout 223
parameters 31
running 41, 128
DRL3nrr record procedure 20
DRL3nrrn record procedure 21
DRL3nrrm record procedure 21
DRLDUMP ddname 40, 128
DRLCHKI ddname 40, 127
DRLICHKO ddname 40, 127
DRLICOMP ddname 40, 127
DRLIPARM ddname 41, 128
DRLIRPTn ddname 40, 128
DRLOUT ddname 40, 127
layout 223
DRLIPARM ddname 41, 128
DRLIRPTn ddname 20, 21, 40, 128
DRLJXIDC assembler DSELECT macro 223
DRLJXIMS archive exit 229
DRLOUT
ddname 40, 127
log procedure report 18, 113
DRLSLOGP
log procedures parameters 121
DSELECT
for log record 219
macro for DRLJXIDC 223
duplicate key condition 42
EMH
composite record section 63
set record group 12
EMH Fast Path record definition synopnt 57

EMH Fast Path record definition (continued)
synopnt failure 57
EMH_BUFFER_BYTES data column 75
EMH_DS_LMSG_BYTES data column 75
EMH_DS_LMSG IRS data column 75
EMH_DS_LMSG IRS data column 75
EMH_DS_LMSG IRS data column 75
EMH_FF_ABORTS data column 75
EMH_FF_ABORTS data column 75
EMH_FF_ABORTS data column 75
EMH_FF_ABORTS data column 75
EMH_FF_ABORTS data column 75
EMH_FP_ABORTS data column 75
EMH_FP_ABORTS data column 75
EMH_FP_ABORTS data column 75
EMH_FP_ABORTS data column 75
EMH_INPUT_SEC data column 75
EMH_MSQQ_OUTPUTS data column 75
EMH_NETWORK_SEC data column 75
EMH_OUTPUT_SEC data column 75
EMH_PROCESS_SEC data column 75
EMH_RESPONSE_SEC data column 75
EMH_TRAN_CNTR_1 data column 75
EMH.TRAN_CNTR_2 data column 75
EMH.TRAN_CNTR_3 data column 75
EMH.TRAN_CNTR_4 data column 75
EMH.TRAN_CNTR_5 data column 75
EMH.TRAN_CNTR_6 data column 75
end of conversation record definition 50
environment information 5, 105
ESS composite record section 62
event accounting record definition 48
extended checkpoint record definition 50
external subsystem
DB2 snap in-doubt record definition 56
record definition 56

F
failure during collect commit 43
Fast Path record definition database DMAC 58
DEDB area data set close 57
DEDB area data set open 57
DEDB area data set set open 57
DEDB area data set status 57
DEDB database open 57
DEDB database update 57
dequeue message 57
hot standby MSDB relocation 58
input message 56
MSDB change 57
output message 57
sequential dependent syncpoint 57
Fast Path special log procedure case 17, 112
FASTPATH parameter 32, 34
FPB_DIR_IO_COUNT data column 93
FPB_DIR_NO_ENTRIES data column 93
FPB_DIR_REQUESTS data column 93
FPB_FBLOCK_IFETCHQ data column 93
FPB_FBLOCKIGATION data column 93
FPB_FBLOCK_REQ data column 93
FPB_FBLOCK_WASHES data column 93

244 Tivoli Decision Support for OS/390: IMS Performance Feature Guide and Reference
Looking at the page, it appears to be a technical document discussing system parameters and metrics related to Tivoli Decision Support for OS/390, specifically focusing on IMS Performance. Here's a breakdown of the content:

**Parameters**
- **MSGQ_FP_COMMITS**
- **MSGQ_FP_ABORTS**
- **MSGQ_FF_COMMITS**
- **MSGQ_FF_ABORTS**
- **MSGQ_DS_SPA_ISRT**
- **MSGQ_DS_SMSG_ISRT**
- **MSGQ_DS_SMSG_BYTES**
- **MSGQ_DS_LMSG_ISRT**
- **MSGQ_DS_LMSG_BYTES**
- **MSGQ_CONV_TRAN**
- **MSGQ_CONV_STARTED**

**M**
- M summarization level suffix
- main pools statistics record definition
- feedback
- online
- ordering
- MAXFREE parameter
- maximizing capabilities of the IMS
- MAXOUTPUT parameter
- message
- root information
- message queue pool detail by date
- report
- message queue record definition
- cancel
- dequeue
- DRRN free
- enqueue
- from CNT
- from PSB or IMS
- prefix changed
- reject
- syncpoint fail
- syncpoint transfer
- message retrieval tool, LookAt
- message-driven BMP program special log procedure
- mode multiple special log procedure
- message multiple special log procedure case
- MSGQ_CONV_STARTED data column
- MSGQ_CONV_TRAN data column
- MSGQ_DS_LMSGgetBytes data column
- MSGQ_DS_LMSG_ISRIT data column
- MSGQ_DS_SMSGgetBytes data column
- MSGQ_DS_SMSG_ISRIT data column
- MSGQ_DS_SPA_ISRIT data column
- MSGQ_FF_ABORTS data column
- MSGQ_FF_COMMITS data column
- MSGQ_FF_COMMITD data column
- MSGQ_FP_ABORTS data column
- MSGQ_FP_COMMITD data column

**N**
- naming standard for tables
- NEW_BLOCK_CREATES data column
- non-message-driven BMP program special log procedure case
- mode multiple special log procedure case
- MSGQ_CONV_STARTED data column
- MSGQ_CONV_TRAN data column
- MSGQ_DS_LMSGgetBytes data column
- MSGQ_DS_LMSG_ISRIT data column
- MSGQ_DS_SMSGgetBytes data column
- MSGQ_DS_SMSG_ISRIT data column
- MSGQ_DS_SPA_ISRIT data column
- MSGQ_FF_ABORTS data column
- MSGQ_FF_COMMITS data column
- MSGQ_FF_COMMITD data column
- MSGQ_FP_ABORTS data column
- MSGQ_FP_COMMITD data column
- MSGQ_INPUT_MSGS data column
- MSGQ_INPUT_MSGS data column
- MSGQ_MSG_SWITCHES data column
- MSGQ_NETWORK_SEC data column
- MSGQ_OUTPUT_MSGS data column
- MSGQ_OUTPUT_SEC data column
- MSGQ_PROCESS_SEC data column
- MSGQ_RESPONSE_SEC data column
- MSGQ_RESPONSES data column
- MSGQ_TRNSIT_SEC data column
- MSGQ_TRAN_CNTR_1 data column
- MSGQ_TRAN_CNTR_2 data column
- MSGQ_TRAN_CNTR_3 data column
- MSGQ_TRAN_CNTR_4 data column
- MSGQ_TRANCTIONS data column
- multiple outputs special log procedure case
- multiple segment input special log procedure case
- multiple segment output special log procedure case
- multiple system coupling special log procedure case
- multiple transactions per schedule of PSB special log procedure case
- MVC_SYSTEM_ID key column
- MVC_SYSTEM_ID parameter

**O**
- OLDS
- padding record
- online publications
- ordering publications
- OSAM/ISAM buffer pool detail by date
- output message reenqueue special log procedure case
- output set (D2) record group
- overview of key columns
- overview of Tivoli Decision Support for OS/390 data flow
- overview report

**P**
- parameter report from log procedure
- parameters
- CICSNAME
- DRL2LOGP
- FASTPATH

These parameters and metrics are crucial for diagnosing and optimizing system performance, ensuring that the IMS Performance is running efficiently.
tabular report 161
terminal message switch special log
procedure case 16, 111
terms defined 241
testing the installation of the IMS
Performance feature 9, 108
TIME key column 66, 68, 69, 80, 83, 85,
87, 89, 90, 92, 96
TIMESTAMP key column 87, 89, 90, 92,
96
title of report 159
trace table log record definition 58
TRANS_TYPE key column 145
transaction abend/completion codes 43
transaction incomplete in composite
record 14
transaction report
application overview graphical 169
application overview tabular 169
application trend graphical 173
application trend tabular 173
region detail by date tabular 191
utilization detail by date tabular 183
volumes for system trend
graphical 177
volumes for system trend
tabular 177
Transaction subcomponent tables 69
transaction type key column
description 81
TRANSACTION_CLASS data
column 70
TRANSACTION_CLASS key column 68,
69, 83
TRANSACTION_NAME data
column 91
TRANSACTION_NAME key column 68,
69, 80, 98
TRANSACTION>Type data column 70
TRANSACTION_TYPE key column 70,
84
trend report definition 160
type of report 160

V
views
for transaction transit time
component 135
IMS_TRAN_QUEUE_QV_DV 147
VSAM buffer pool detail by date
report 188
VSAM subpool statistics record
definition 55
VSAM_ESDS_INSERTS data column 97
VSAM_GET_CALLS data column 97
VSAM_KSDS_INSERTS data column 97
VSAM_NUSER_Writes data
column 97
VSAM_READS data column 97
VSAM_SCHBFR_CALLS data
column 97
VSAM_USER_Writes data column 97

W
W summarization level suffix 65
wait for input program special log
procedure case 17, 112
WAITS_NO_BUFFER data column 88
WAITS_RLSE_OWN data column 88
worst case report 160
WRITEPENDING parameter 33, 37, 122,
124
AVES_BFR_STEAL data column 88

X
XRFNAME parameter 33, 37, 122, 124
XRF_SYS parameter 33, 37, 122, 124

U
unit-of-recovery set record group 12
updating lookup tables
definition 8
IMS_APPLICATION 8
non-IMS Performance feature lookup
tables 9, 108
user ID report
resource worst case by date 193
response time and CPU detail by
date 181
USER_ID key column 68, 80, 86, 99
utilization report
program worst case by date 194
region by date detail 190
transaction by date detail 183