Tivoli® Decision Support for z/OS®

IMS™ Performance Feature Guide and Reference

Version 1.7
Note

Before using this information and the product it supports, read the information in “Notices” on page 255.
# Contents

**Figures** ............................... vii  
**Tables** ............................... ix  

**Preface** ............................... xi  
Who should read this book xi  
What this book contains xi  
Publications xii  
Tivoli Decision Support for z/OS library xii  
Using LookAt to look up message explanations xiii  
Accessing publications online xiv  
Ordering publications xiv  
Accessibility xiv  
Tivoli technical training xiv  
Contacting IBM Software Support xiv  
Determine the business impact of your problem xv  
Describe your problem and gather background information xvi  
Submit your problem to IBM Software Support xvi  
Search knowledge bases xvi  
  Search the information center on your local system or network xvi  
  Search the Internet xvi  
Obtaining fixes xvii  
Updating support information xvii  
Conventions used in this book xviii  
Typeface conventions xviii  
Changes in this edition xviii  

**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 z/OS 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 z/OS 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  
  Running the IMS Light feature 42  
  Setting up Load Library for the IMS Light feature 43  
    Statement description 43  
  Setting up the Tivoli Decision Support collect for the IMS Light feature 44  
  Recovering from abends during collect 45  
    Recovery using the log procedure checkpoint facility 45  
    Recovery without the checkpoint facility 45  
  Additional capabilities 45  

**Chapter 5. IMS performance feature log and record definitions** ............................... 47  
Log definitions 47  
Record definitions 48  
  Comparison of performance programs 48  
  Descriptions of record definitions 49  
Composite record definitions 61  
  Composite record sections in IMS_Vnnn_TRAN 61  
  Composite record types and subtypes in IMS_Vnnn_TRAN 66
Chapter 6. IMS performance feature
data tables and lookup tables ..... 67
Naming standard for table ..... 67
Table descriptions ..... 67
Control tables ..... 68
IMS log records component data tables ..... 68
IMS collect component data tables ..... 68
Table and key column cross-reference ..... 70
Transaction subcomponent tables ..... 71
IMS_TRANSACTION_H_.D_.W ..... 71
IMS_USER_TRAN_H_.D_.W ..... 82
IMS_TRAN_TYPE key column ..... 83
System subcomponent tables ..... 85
IMS_SYSTEM_Q_.D. ..... 85
Application subcomponent tables ..... 87
IMS_APPLICATION_H_.W ..... 87
IMS_USER_APPD_.W. ..... 88
Statistics subcomponent tables ..... 89
IMS_CHKPT_IOSAM_T ..... 89
IMS_CHKPT_POOL_T ..... 91
IMS_CHKPT_REGION_T ..... 92
IMS_CHKPT_STATS_T ..... 94
IMS_CHKPT_VSAM_T ..... 98
Lookups ..... 100
IMS_APPLICATION ..... 100
Example of table contents ..... 100
Using the GROUP_ID lookup table ..... 101
GROUP_ID lookup table ..... 101

Part 2. IMS Shared Queue feature ..... 103

Chapter 7. Introduction to the IMS
Shared Queue feature. ..... 105
Understanding the IMS Shared Queue feature ..... 106
Collecting data ..... 106
SLDS ..... 107
Log procedure ..... 107
Composite record ..... 107
Record procedure ..... 107
Record definitions ..... 107
Tivoli Decision Support for z/OS data tables and environmental information ..... 107
Report ..... 107
The log collector and DRLSLOGP ..... 107
Installing and customizing the IMS Shared Queue feature ..... 108
Planning for the IMS Shared Queue feature ..... 109
Selecting IMS Shared Queue feature components ..... 109
The collect components ..... 110
The log records component. ..... 110
Updating other lookup and control tables ..... 110
Using the IMS Shared Queue feature ..... 111

Chapter 8. Using log and record
procedures within the IMS Shared Queue ..... 113
The log procedure ..... 113
Set relationships ..... 113
Composite records and subtypes ..... 114
Handling of special IMS cases within the IMS
Shared Queue ..... 114
Log procedure DRLOUT reports ..... 117

Chapter 9. Understanding data flow
through IMS Shared Queue. ..... 119
Log collector data flow ..... 120
DRLSLOGP data flow ..... 121
IMS Shared Queue Collect feature data flow ..... 122

Chapter 10. Administering the IMS
Shared Queue feature. ..... 125
IMS Shared Queue Logs ..... 125
Specifying DRLSLOGP and log procedure parameters ..... 125
Specifying the log collector parameter ..... 128
Running the log collector in a shared queue environment ..... 129
Using DRLSLOGP ..... 130
DRLSLOGP input and output data sets ..... 130
Running DRLSLOGP ..... 131
Operational considerations ..... 132
Running the IMS Light feature ..... 133
Setting up Load Library for the IMS Light feature ..... 134
Statement description ..... 134
Setting up the Tivoli Decision Support collect for the IMS Light feature ..... 135

Chapter 11. IMS Shared Queue record
definitions. ..... 137
Composite record definitions ..... 137
Composite record sections ..... 137

Chapter 12. IMS Shared Queue data
tables and lookup tables. ..... 139
IMS Shared Queue data tables. ..... 139
Transaction Transit Time subcomponent tables and views ..... 139
Data tables ..... 140
IMS.TRAN_H_.D_.W ..... 140
IMS.TRAN_QVQ_.D. ..... 144
IMS_SYSTEM.TRAN_.H_.D. ..... 147
TRANS_TYPE key column ..... 149
Views ..... 151
IMS.TRAN_QVQ_.DV ..... 151
Account and Availability subcomponent tables ..... 152
Data Tables ..... 152
IMS.PSB.ACCOUNT_.H_.D_.W ..... 152
Availability for IMS resources ..... 157
IMS_AVAILABILITY_T ..... 158
IMS_AVAILABILITY_D_.W. ..... 159
Lookup tables ..... 160
IMS_AVAIL_RESOURCE ..... 160
Example of Table Contents ..... 160
Mapping between Non-SQ and SQ DB2 Tables ..... 161
IMS.TRANSACTION_.H_.D_.W. ..... 161
IMS_USER.TRAN_.x_.(H_.D_.W) ..... 168
IMS_SYSTEM.x_.(Q_.D). ..... 169
IMS_APPLICATION_x (H, W) .......................... 169
IMS_USER_APPL_x (D, W) .......................... 169
Mapping between New DB2 Table Fields and CSQ
Records .............................................. 169
IMS_TRAN_x (H, D, W) .............................. 169
IMS_TRAN_QUEUE_x (Q, D) ......................... 170
IMS_SYSTEM_TRAN_x (H, D) ......................... 170
IMS_PSB_ACCOUNT_x (H, D, W) .................... 171
New IMS CSQ Composite Record Header Layout
(DRLCSQCR) ........................................ 171
Mapping between SQ and non-SQ IMS R2 Fields 172

Part 3. Appendices .................................. 175

Appendix A. Reports ................................. 177
Report format and general description .............. 177
Report title ......................................... 177
Report ID ............................................ 177
Report group ........................................ 178
Source .............................................. 178
Attributes .......................................... 178
Variables ............................................ 178
Report types ........................................ 178
Standard report formats ............................... 179
Tabular reports ...................................... 179
Graphic reports ...................................... 179
Samples of reports across non-SQ and SQ tables 180
IMS overview reports ................................ 185
IMS Application Response Time Overview report 185
IMS Application Transaction Overview report 187
IMS trend reports .................................... 189
IMS Application Response Time Trend report 189
IMS Application Transaction Trend report .......... 191
IMS Application CPU Utilization Trend report 192
IMS System Response Time Trend report .......... 193
IMS System System Transaction Volumes Trend report 195
IMS System CPU Utilization Trend report 197
IMS System DLI Utilization Trend report 198
IMS detail reports ................................... 199
IMS User ID Response Time and CPU Detail by
Date report .......................................... 199
IMS Transaction Utilization Detail by Date report 199
IMS Message Queue Pool Detail by Date report 201
IMS OSAM/ISAM Buffer Pool Detail by Date report 204
IMS VSAM Buffer Pool Detail by Date report 206
IMS Region Utilization Detail by Date report 208
IMS Region Detail by Date report 209
IMS Resource Detail by Quarter Hour report 210
IMS worst case reports .............................. 211
IMS User ID Resource Worst Case by Date
report ................................................. 211
IMS Program Utilization Worst Case by Date
report ................................................. 212
IMS Availability reports ............................. 213
IMS CSQ Subsystem Availability, Daily Trend
Report .................................................. 213
IMS CSQ Region Availability, Daily Overview
report .................................................. 215
IMS CSQ Application Usage and Availability
report ................................................... 216
IMS Message Queue Reports ....................... 218
IMS Message Queue Utilization, Date report 219
IMS Message Queue Utilization by Transaction, Date report 221
IMS Message Queue Utilization Overview, Daily
Report .................................................. 222
IMS Transaction Arrival Rate and Message
Queue Usage, Daily Trend ......................... 223
IMS CSQ Transaction Transit Time Reports 224
IMS CSQ Transaction Time Analysis By Transaction
Name ................................................... 224
IMS CSQ Transaction Time Analysis By LTERM and
Userid .................................................. 227
IMS CSQ Transaction Time Analysis by Region 231
IMS CSQ Utilization Reports ........................ 234
IMS CSQ Resource Utilization, Daily Overview 234

Appendix B. Creating IMS log record
DSECTs .............................................. 237

Appendix C. DRLJXIDC DSECT macro 241

Appendix D. Sample archive exit .................. 247

Appendix E. DFSLTMG0 log merge
utility .................................................. 249
Controlling the log merge ........................... 250

Appendix F. List of abbreviations ............... 251

Notices .............................................. 255
Trademarks ......................................... 257

Glossary ............................................. 259

Index ............................................... 261

Contents V
Figures

1. Overview of data collection using the IMS Performance feature ........................................ 4
2. Task flow for Tivoli Decision Support for z/OS features .................................................... 6
3. IMS_APPLICATION lookup table example ....................................................................... 9
4. Example of log procedure parameter report ..................................................................... 18
5. Example of log procedure pending node report ............................................................... 19
6. Example of log procedure composite record report ......................................................... 19
7. Example of R0 report ........................................................................................................ 20
8. Example of R1 report ........................................................................................................ 21
9. Overview of Tivoli Decision Support for z/OS data flow ................................................. 24
10. Data flow through the log collector .................................................................................. 26
11. Data flow through DRL2LOGP ....................................................................................... 27
12. Data flow—statistics subcomponent ............................................................................... 28
13. Data flow—application, transaction, and system subcomponents ..................................... 29
14. Sample job for running the log collector ........................................................................ 39
15. Sample data set with log collector parameters ................................................................ 39
16. Sample job for running DRL2LOGP ................................................................................. 41
17. Key columns in transaction, system, and application subcomponent tables ...................... 70
18. Overview of data collection using the IMS Shared Queue feature .................................. 106
19. Task flow for Tivoli Decision Support for z/OS features ................................................ 108
20. Example of log procedure parameter report within IMS Shared Queue ......................... 118
21. Example of log procedure pending node report within IMS Shared Queue .................... 118
22. IMS Shared Queue data flow through the log collector .................................................... 120
23. Data flow through DRLSLOGP ....................................................................................... 121
24. Data flow—Account and Availability subcomponent ...................................................... 122
25. Data flow—Transaction Transit Time subcomponent ...................................................... 123
26. Sample job for running the log collector within IMS Shared Queue ................................ 129
27. Sample jobs for running DRLSLOGP ............................................................................. 132
28. Tabular reports example ................................................................................................. 179
29. Graphic reports example ............................................................................................... 180
30. IMSY01 Report Query (DRLQIY01 member) ................................................................. 181
31. Example of Query from IMS_SYSTEM_tran_D Shared Queue Table ................................. 182
32. Example of Mixed Query ............................................................................................... 183
33. Example of Query Input Variables Panel DRLDSEL .................................................... 184
34. Example of IMS Application Response Time Overview graphic report ......................... 185
35. Example of IMS Application Transaction Overview graphic report ................................ 187
36. Example of IMS Application Response Time Trend graphic report .............................. 189
37. Example of IMS Application Transaction Trend graphic report .................................... 191
38. Example of IMS Application CPU Utilization Trend graphic report .......................... 192
39. Example of IMS System Response Time Trend graphic report ...................................... 193
40. Example of IMS System Transaction Volumes Trend graphic report .......................... 195
41. Example of IMS System CPU Utilization Trend graphic report .................................... 197
42. Example of IMS System DLI Utilization Trend graphic report ........................................ 198
43. Example of IMS User ID Response Time and CPU Detail by Date tabular report ........... 199
44. Example of IMS Transaction Utilization Detail by Date tabular report .......................... 201
45. Example of IMS Message Queue Pool Detail by Date tabular report ............................ 202
46. Example of IMS OSAM/ISAM Buffer Pool Detail by Date tabular report ...................... 204
47. Example of IMS VSAM Buffer Pool Detail by Date tabular report ................................... 206
48. Example of IMS Region Utilization Detail by Date tabular report ................................... 208
49. Example of IMS Region Detail by Date tabular report .................................................. 209
50. Example of IMS Resource Detail by Quarter Hour tabular report ............................... 210
51. Example of IMS User ID Resource Worst Case by Date tabular report .......................... 211
52. Example of IMS Program Utilization Worst Case by Date tabular report ...................... 212
53. Example of an IMS CSQ subsystem Availability, Daily Trend Report ........................... 213
54. Example of an IMS CSQ Regain Availability, Daily Overview report ............................ 215
55. Example of an IMS CSQ application Usage and Availability report ............................... 216
56. Example of IMS Message Queue Utilization, Date report ............................................. 219
57. Example of IMS Message Queue Utilization by Transaction, Date report ..................... 221
58. Example of an IMS Message Queue Utilization Overview, Daily Report ...................... 222
59. Example of an IMS Transaction Arrival Rate and Message Queue Usage, Daily Trend .... 223
60. Example of IMS CSQ Transit Time Analysis By Transaction Name, Daily report .......... 224
61. Example of IMS CSQ Transit Time Analysis By LTERM and Userid, Daily report .......... 228
62. Example of IMS CSQ Transit Time Analysis By Region, Daily report ............................ 231
63. Example of IMS CSQ Resource Utilization, Daily Overview .......................................... 235
64. Sample JCL for assembling IMS log record DSECTs ..................................................... 238
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>65.</td>
<td>DRLJXIDC MACRO . . . . . . . . 242</td>
</tr>
<tr>
<td>66.</td>
<td>Sample IMS archive exit . . . . . . . 248</td>
</tr>
<tr>
<td>67.</td>
<td>DFSLTMG0 log merge utility . . . . . . 249</td>
</tr>
<tr>
<td></td>
<td>Tables</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>Logical sets for a full function transaction</td>
</tr>
<tr>
<td>2</td>
<td>Main record set combinations</td>
</tr>
<tr>
<td>3</td>
<td>Parameter summary for DRL2LOGP</td>
</tr>
<tr>
<td>4</td>
<td>Parameter summary for the log procedure</td>
</tr>
<tr>
<td>5</td>
<td>Comparison of the IMS Performance feature with other products for IMS</td>
</tr>
<tr>
<td>6</td>
<td>IMS record types and IMS Performance feature record definitions</td>
</tr>
<tr>
<td>7</td>
<td>IMS_Vnnum_TRAN</td>
</tr>
<tr>
<td>8</td>
<td>Composite record types and subtype sections in IMS_Vnnum_TRAN</td>
</tr>
<tr>
<td>9</td>
<td>GROUP_ID lookup table</td>
</tr>
<tr>
<td>10</td>
<td>Parameter summary for DRLSLOGP</td>
</tr>
<tr>
<td>11</td>
<td>Parameter summary for the log procedure within IMS Shared Queue</td>
</tr>
<tr>
<td>12</td>
<td>Composite record sections</td>
</tr>
<tr>
<td>13</td>
<td>IMS_TRANSACTION_H, D, W Field Remap</td>
</tr>
<tr>
<td>14</td>
<td>IMS_USER_TRAN_x (H, D, W) Field Remap</td>
</tr>
<tr>
<td>15</td>
<td>IMS_SYSTEM_x (Q, D) Field Remap</td>
</tr>
<tr>
<td>16</td>
<td>IMS_TRAN_x (H, D, W) Description</td>
</tr>
<tr>
<td>17</td>
<td>IMS_TRAN_QUEUE_x (Q, D) Description</td>
</tr>
<tr>
<td>18</td>
<td>IMS_SYSTEM_TRAN_x (H, D) Description</td>
</tr>
<tr>
<td>19</td>
<td>IMS_PSB_ACCOUNT_x (H, D, W) Description</td>
</tr>
<tr>
<td>20</td>
<td>SQ and non-SQ IMS R2 Field Remap</td>
</tr>
</tbody>
</table>
Preface

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

This book allows you to install and use the IMS Performance and IMS Shared Queue features of Tivoli Decision Support for z/OS. 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 z/OS reports, so you can analyze the characteristics of your system

Tivoli Decision Support for z/OS was previously known as Tivoli Decision Support for OS/390®.

The following terms are used interchangeably throughout this book:

• Tivoli Decision Support for z/OS and Tivoli Decision Support for OS/390
• MVS™, OS/390, and z/OS

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 z/OS administrators (primarily as a reference to table and report definitions)
• Users with various backgrounds who are interested in analyzing IMS performance data and improving IMS performance

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

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”
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, “DFSLETMG0 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 z/OS 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 z/OS library

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

- Accounting Feature for z/OS, SH19-4495
  Provides information for users who want to use Tivoli Decision Support for z/OS to collect and report performance data generated by the Accounting Feature for z/OS.
- Administration Guide, SH19-6816
  Provides information about initializing the Tivoli Decision Support for z/OS database and customizing and administering Tivoli Decision Support for z/OS.
- AS/400 System Performance Feature Guide and Reference, SH19-4019
  Provides information for administrators and users about collecting and reporting performance data generated by AS/400® systems.
- CICS Performance Feature Guide and Reference, SH19-6820
  Provides information for administrators and users about collecting and reporting performance data generated by Customer Information and Control System (CICS®).
- Distributed Systems Performance Feature Guide and Reference, SH19-4018
  Provides information for administrators and users about collecting and reporting performance data generated by operating systems and applications running on a workstation.
- Guide to 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.

- **IMS Performance Feature Guide and Reference, SH19-6825**  
  Provides information for administrators and users about collecting and reporting performance data generated by Information Management System (IMS).

- **Language Guide and Reference, SH19-6817**  
  Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.

- **Messages and Problem Determination, SH19-6902**  
  Provides information to help operators and system programmers understand, interpret, and respond to Tivoli Decision Support for z/OS messages and codes.

- **Network Performance Feature 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 z/OS 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 z/OS to analyze Multiple Virtual Storage (MVS) or Virtual Machine (VM) performance data.

- **IBM Online Library z/OS Software Products Collection Kit, SK3T-4270**  
  CD containing all z/OS 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/](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. Access the Tivoli software information center by first going to the Tivoli software library at the following Web address:

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

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

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

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 the Administration Guide.

Tivoli technical training

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

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

Contacting IBM Software Support

IBM Software Support provides assistance with product defects.
Before contacting IBM Software Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have:

- For IBM distributed software products (including, but not limited to, Tivoli, Lotus®, and Rational® products, as well as DB2® and WebSphere® products that run on Windows® or UNIX operating systems), enroll in Passport Advantage® in one of the following ways:
  - **Online**: Go to the Passport Advantage® Web page [http://www.lotus.com/services/passport.nsf/WebDocs/Passport_Advantage_Home] and click How to Enroll.
  - **By phone**: For the phone number to call in your country, go to the IBM Software Support Web site [http://techsupport.services.ibm.com/guides/contacts.html] and click the name of your geographic region.
- For IBM eServer™ software products (including, but not limited to, DB2 and WebSphere products that run in zSeries®, pSeries®, and iSeries® environments), you can purchase a software maintenance agreement by working directly with an IBM sales representative or an IBM Business Partner. For more information about support for eServer software products, go to the IBM Technical Support Advantage Web page [http://www.ibm.com/servers/eserver/techsupport.html].

If you are not sure what type of software maintenance contract you need, call 1-800-IBMSERV (1-800-426-7378) in the United States or, from other countries, go to the contacts page of the IBM Software Support Handbook on the Web [http://techsupport.services.ibm.com/guides/contacts.html] and click the name of your geographic region for phone numbers of people who provide support for your location.

Follow the steps in this topic to contact IBM Software Support:

1. “Determine the business impact of your problem”
2. “Describe your problem and gather background information” on page xvi
3. “Submit your problem to IBM Software Support” on page xvi

### Determine the business impact of your problem

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem you are reporting. Use the following criteria:

<table>
<thead>
<tr>
<th>Severity 1</th>
<th>Critical business impact: You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Severity 2</td>
<td>Significant business impact: The program is usable but is severely limited.</td>
</tr>
<tr>
<td>Severity 3</td>
<td>Some business impact: The program is usable with less significant features (not critical to operations) unavailable.</td>
</tr>
<tr>
<td>Severity 4</td>
<td>Minimal business impact: The problem causes little impact on operations, or a reasonable circumvention to the problem has been implemented.</td>
</tr>
</tbody>
</table>
Contacting IBM Software Support

Describe your problem and gather background information

When explaining a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Software Support specialists can help you solve the problem efficiently. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Software Support is likely to ask for this information.
- Can the problem be recreated? If so, what steps led to the failure?
- Have any changes been made to the system? (For example, hardware, operating system, networking software, and so on.)
- Are you currently using a workaround for this problem? If so, please be prepared to explain it when you report the problem.

Submit your problem to IBM Software Support

You can submit your problem in one of two ways:

- **By phone**: For the phone number to call in your country, go to the contacts page of the IBM Software Support Handbook on the Web [http://techsupport.services.ibm.com/guides/contacts.html](http://techsupport.services.ibm.com/guides/contacts.html) and click the name of your geographic region.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Software Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Software Support provides a workaround for you to implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM product support Web pages daily, so that other users who experience the same problem can benefit from the same resolutions.

For more information about problem resolution, see "Searching knowledge bases" and "Obtaining fixes" on page xvii.

Searching knowledge bases

If you have a problem with your IBM software, you want it resolved quickly. Begin by searching the available knowledge bases to determine whether the resolution to your problem is already documented.

Search the information center on your local system or network

IBM provides extensive documentation that can be installed on your local machine or on an intranet server. You can use the search function of this information center to query conceptual information, instructions for completing tasks, reference information, and support documents.

Search the Internet

If you cannot find an answer to your question in the information center, search the Internet for the latest, most complete information that might help you resolve your problem. To search multiple Internet resources for your product, expand the product folder in the navigation frame to the left and select **Support on the Web**. From this topic, you can search a variety of resources including:

- IBM technotes
• IBM downloads
• IBM Redbooks™
• IBM DeveloperWorks
• Forums and newsgroups
• Google

Obtaining fixes
A product fix might be available to resolve your problem. You can determine what fixes are available for your IBM software product by checking the product support Web site:
2. Under Products A - Z, select your product name. This opens a product-specific support site.
3. Under Self help, follow the link to All Updates, where you will find a list of fixes, fix packs, and other service updates for your product. For tips on refining your search, click Search tips.
4. Click the name of a fix to read the description and optionally download the fix.
To receive weekly e-mail notifications about fixes and other news about IBM products, follow these steps:
1. From the support page for any IBM product, click My support in the upper-right corner of the page.
2. If you have already registered, skip to the next step. If you have not registered, click register in the upper-right corner of the support page to establish your user ID and password.
3. Sign in to My support.
4. On the My support page, click Edit profiles in the left navigation pane, and scroll to Select Mail Preferences. Select a product family and check the appropriate boxes for the type of information you want.
5. Click Submit.
6. For e-mail notification for other products, repeat Steps 4 and 5.

Updating support information
Information centers typically include one or more support information plug-ins. These plug-ins add IBM technotes and other support documents to the information center. The following steps describe how to update your support information plug-ins:
2. Under Products A - Z, select your product name. This opens a product-specific support site.
3. Under Search support for this product, type the keyword phrase: com.ibm.support. Click the Download check box, and click Submit.
4. Check the search results for updates to support information plug-ins. All support information plug-ins follow the naming convention, "com.ibm.support.product.doc." If an update is available, select it from the list and view the download instructions.
5. Save the attached zip file to a temporary location on your hard drive.
6. Unzip the downloaded file, making sure that you retain the subfolders.
7. From the location where you unzipped the file, copy the support information plug-in folder to your Eclipse plug-ins folder. For example, if your IBM software product is installed at `c:\IBM\WebSphere\`, copy the updated plug-in folder (com.ibm.support.product.doc) to `c:\IBM\WebSphere\eclipse\plugins`.
8. To see the updated support information, start the information center (or shut it down and restart it), and expand the Support information node in the navigation tree.

### 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 terms MVS, OS/390, and z/OS are used interchangeably throughout this book.

#### 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, multicolon lists, containers, menu choices, menu names, tabs, property sheets), labels (such as Tip:, and Operating system considerations)
- Column headings in a table
- Keywords and parameters in text

**Italic**

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

**Monospace**

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

### Changes in this edition

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

- The name of the product has been changed to Tivoli Decision Support for z/OS (except in figures).
Contacting IBM Software Support

- References to IMS Version 5.1 and 6.1 have been removed, because no longer supported.
- Some fields of the data table "IMS_CHKPT_VSAM_T" on page 98 have been updated.

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

Chapter 1. Introduction to the IMS Performance feature
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 z/OS 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 ...
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
Overview of Tivoli Decision Support for z/OS 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
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
Running the IMS Light feature ............................................................... 42
Setting up Load Library for the IMS Light feature ................................. 42
Statement description ............................................................................ 43
Setting up the Tivoli Decision Support collect for the IMS Light feature 44
Recovering from abends during collect ................................................... 45
Recovery using the log procedure checkpoint facility ............................ 45
Recovery without the checkpoint facility .............................................. 45
Additional capabilities .......................................................................... 45

Chapter 5. IMS performance feature log and record definitions............. 47
Log definitions ....................................................................................... 47
Record definitions .................................................................................. 48
Comparison of performance programs .................................................. 48
Descriptions of record definitions ......................................................... 49
Composite record definitions ................................................................. 61
Composite record sections in IMS_Vnnn_TRAN .................................. 61
Composite record types and subtypes in IMS_Vnnn_TRAN ..................... 66

Chapter 6. IMS performance feature data tables and lookup tables .......... 67
Naming standard for tables .................................................................... 67
Table descriptions .................................................................................. 67
Control tables ....................................................................................... 68
IMS log records component data tables ................................................. 68
IMS collect component data tables ....................................................... 68
Table and key column cross-reference .................................................. 70
Transaction subcomponent tables ......................................................... 71
IMS.TRANSACTION.H._D._W ............................................................... 71
IMS_USER.TRAN.H._D._W ................................................................. 82
IMS.TRAN.TYPE key column .............................................................. 83
System subcomponent tables ............................................................... 85
IMS.SYSTEM.Q._D ............................................................................. 85
Application subcomponent tables ......................................................... 87
IMS.APPLICATION.H._W ................................................................. 87
IMS_USER.APPL.D._W ................................................................. 88
Statistics subcomponent tables ............................................................ 89
IMS_CHKPT_IOSAM_T ................................................................. 89
IMS_CHKPT_POOLS_T ................................................................. 91
IMS_CHKPT_REGION_T ................................................................. 92
IMS_CHKPT_STATS_T ................................................................. 94
IMS_CHKPT_VSAM_T ................................................................. 98
Lookup tables ....................................................................................... 100
IMS_APPLICATION ................................................................. 100
Example of table contents ................................................................. 100
Using the GROUP_ID lookup table ....................................................... 101
GROUP_ID lookup table ................................................................. 101

Introductory paragraph here
Chapter 1. Introduction to the IMS Performance feature

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

The Tivoli Decision Support for z/OS base includes:

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

Each feature provides:

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

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

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 z/OS 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 z/OS 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 z/OS 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 z/OS record procedure. The record definitions are used by
Tivoli Decision Support for z/OS collect when updating DB2 tables.

Tivoli Decision Support for z/OS data tables and environmental
information
The IMS Performance feature uses the R2 records, along with user-supplied data in
Tivoli Decision Support for z/OS 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 z/OS 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 z/OS log collector program. The log collector uses record definitions
and other definitions when it updates Tivoli Decision Support for z/OS data
tables. The log collector is part of the Tivoli Decision Support for z/OS 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 z/OS feature component.

To install and use the IMS Performance feature, you must have an MVS operating system capable of running Tivoli Decision Support for z/OS. The IMS Performance feature supports data from systems running IMS/ESA 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 z/OS 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 z/OS 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 z/OS 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 z/OS collects unnecessary data, which takes up disk space.

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

The IMS Performance feature components are:
- 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 times, system transaction volumes, CPU and database utilization, and transaction detail.

**System subcomponent** Collects information about general system activity.
Introduction to the IMS Performance Feature

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 z/OS record definitions for IMS records in the SLDS. You can write your own Tivoli Decision Support for z/OS 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 47 for descriptions of the Tivoli Decision Support for z/OS 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 z/OS 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 z/OS 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 z/OS 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 z/OS 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 z/OS. For specific information about running the IMS Performance feature, see .

<table>
<thead>
<tr>
<th>TRANSACTION_NAME</th>
<th>PROGRAM_NAME</th>
<th>APPLICATION_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE01%</td>
<td>E01%</td>
<td>GENERAL LEDGER</td>
</tr>
<tr>
<td>VA%</td>
<td>A%</td>
<td>PAYROLL</td>
</tr>
<tr>
<td>VE%</td>
<td>EP%</td>
<td>PENSIONS</td>
</tr>
<tr>
<td>VP%</td>
<td>ER%</td>
<td>PERSONNEL</td>
</tr>
<tr>
<td>TA01</td>
<td>A0100000</td>
<td>ACCOUNT ENQUIRY</td>
</tr>
<tr>
<td>DM%</td>
<td>M0302%</td>
<td>DIRECT MARKETING</td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>OTHER</td>
</tr>
</tbody>
</table>
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 dequeueing (X'36' Out) on a CNT in MSGQ.
Using log and record procedures within IMS

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></td>
<td></td>
<td>X'07'</td>
</tr>
<tr>
<td>IMS issues GU for message</td>
<td></td>
<td></td>
<td>X'31'</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**
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
Using log and record procedures within IMS

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' 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
Using log and record procedures within IMS

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRLIPARM</td>
<td>Reports requested are</td>
</tr>
<tr>
<td>DRLIPARM</td>
<td>Reports requested are</td>
</tr>
<tr>
<td>DRLIPARM</td>
<td>Reports requested are</td>
</tr>
<tr>
<td>DRLIPARM</td>
<td>IMS version</td>
</tr>
</tbody>
</table>

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 235959F |
| Default  | IMSID       | IMP       |
| Default  | RECETYPE    | FF        |
| DRLIPARM | IMSIDCHECK  | CONTINUE   |
| DRLIPARM | WRITEPENDING| YES       |
| Default  | PASSLOGREC  | YES       |
| Default  | FULLFUNC ESS SUPPRESS | NO |
| Default  | FULLFUNC MSGSWSHIT SUPPRESS | NO |
| Default  | FULLFUNC BMP SUPPRESS | NO |
| Default  | FULLFUNC FPSYNC SUPPRESS | NO |
| Default  | FULLFUNC CONV SUPPRESS | NO |
| Default  | FULLFUNC ESS 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 IMS started at 0903242F 09514236 has switched OLDS at 0903242F 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.
Using log and record procedures within IMS

DRL2072I Statistics for NODEs created this run:

<table>
<thead>
<tr>
<th>NODE type</th>
<th>LENGTH</th>
<th>NODE allocated</th>
<th>NODEs pending</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1D1</td>
<td>12</td>
<td>480</td>
<td>480</td>
</tr>
<tr>
<td>D1</td>
<td>76</td>
<td>30400</td>
<td>30400</td>
</tr>
<tr>
<td>D2</td>
<td>40</td>
<td>32000</td>
<td>32000</td>
</tr>
<tr>
<td>EMH-NODE</td>
<td>24</td>
<td>9600</td>
<td>9600</td>
</tr>
<tr>
<td>EMH-RTKN</td>
<td>46</td>
<td>19200</td>
<td>19200</td>
</tr>
<tr>
<td>IMS</td>
<td>80</td>
<td>1280000</td>
<td>1280000</td>
</tr>
<tr>
<td>RTKN</td>
<td>56</td>
<td>22400</td>
<td>22400</td>
</tr>
<tr>
<td>SPA</td>
<td>20</td>
<td>800</td>
<td>800</td>
</tr>
<tr>
<td>UOR</td>
<td>68</td>
<td>136000</td>
<td>136000</td>
</tr>
<tr>
<td>TOTALS</td>
<td>424</td>
<td>1530880</td>
<td>1530880</td>
</tr>
</tbody>
</table>

Figure 5. Example of log procedure pending node report

The log procedure composite record report 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.

DRL2073I Statistics for composite records created this run:

<table>
<thead>
<tr>
<th>Record</th>
<th>Count</th>
<th>Min</th>
<th>Max</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0C Stray UOR with no MsgQ output</td>
<td>37</td>
<td>596</td>
<td>660</td>
<td>24356</td>
</tr>
<tr>
<td>0D Stray UOR with MsgQ output</td>
<td>4</td>
<td>1106</td>
<td>2686</td>
<td>8714</td>
</tr>
<tr>
<td>0F Incomplete composite</td>
<td>112</td>
<td>342</td>
<td>1948</td>
<td>46744</td>
</tr>
<tr>
<td>40 IMS Checkpoint record</td>
<td>1</td>
<td>633</td>
<td>633</td>
<td>633</td>
</tr>
<tr>
<td>45 Checkpoint statistics</td>
<td>15</td>
<td>35</td>
<td>1595</td>
<td>3035</td>
</tr>
<tr>
<td>47 Checkpoint Region active</td>
<td>1</td>
<td>47</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>5A Message switch (incl. MSG/ISC)</td>
<td>2</td>
<td>540</td>
<td>540</td>
<td>1080</td>
</tr>
<tr>
<td>FB System/Program output (incl. MTO traf)</td>
<td>101</td>
<td>490</td>
<td>498</td>
<td>50194</td>
</tr>
<tr>
<td>FE Transaction with no MsgQ output</td>
<td>13</td>
<td>960</td>
<td>1274</td>
<td>15808</td>
</tr>
<tr>
<td>FF Transaction with MsgQ output</td>
<td>157</td>
<td>832</td>
<td>2982</td>
<td>270546</td>
</tr>
</tbody>
</table>

DRL2092I IMS log processing stopped at 08093242F 1042035F.
DRL2093I Read 9063 records from DRLLOG.

Figure 6. Example of log procedure composite record report

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 237), and the DRLXIDC macro supplied with the IMS Performance feature (see Appendix C, “DRLXIDC DSECT macro,” on page 241).

The record procedures provided with the IMS Performance feature are described below.
Using log and record procedures within IMS

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.

DRL3Intr0 (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></td>
<td>Lth</td>
<td>Lth</td>
<td>Bytes</td>
</tr>
<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 SMR 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>248</td>
</tr>
<tr>
<td>08 Application program schedule</td>
<td>162</td>
<td>88</td>
<td>88</td>
</tr>
<tr>
<td>20 Data base open</td>
<td>8</td>
<td>54</td>
<td>54</td>
</tr>
<tr>
<td>31 Message queue Get Unique</td>
<td>668</td>
<td>46</td>
<td>4122</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>88</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>72</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>56</td>
</tr>
<tr>
<td>03 Message XFER record</td>
<td>30</td>
<td>68</td>
<td>72</td>
</tr>
<tr>
<td>08 Message XFER record</td>
<td>24</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>30 COMMIT record</td>
<td>202</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>38 COMMIT record</td>
<td>37</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>40 System checkpoint</td>
<td>487</td>
<td>45</td>
<td>1008</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>936</td>
</tr>
<tr>
<td>04 SMR blocks</td>
<td>263</td>
<td>336</td>
<td>960</td>
</tr>
<tr>
<td>05 CTB blocks</td>
<td>5</td>
<td>648</td>
<td>960</td>
</tr>
<tr>
<td>06 DMB blocks</td>
<td>140</td>
<td>134</td>
<td>552</td>
</tr>
<tr>
<td>07 PSB blocks</td>
<td>29</td>
<td>288</td>
<td>1008</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>940</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>904</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 EOV reached</td>
<td>2</td>
<td>492</td>
<td>492</td>
</tr>
<tr>
<td>43 Log dataset control record</td>
<td>2</td>
<td>210</td>
<td>210</td>
</tr>
<tr>
<td>45 Checkpoint statistics record</td>
<td>15</td>
<td>16</td>
<td>1576</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>46</td>
</tr>
<tr>
<td>4C Data base started/stopped</td>
<td>9</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td>50 Data base update/replace/delete</td>
<td>3148</td>
<td>111</td>
<td>1834</td>
</tr>
<tr>
<td>56 External subsystem TPCC</td>
<td>483</td>
<td>64</td>
<td>72</td>
</tr>
<tr>
<td>00 IMS/VS V1</td>
<td>6</td>
<td>72</td>
<td>72</td>
</tr>
<tr>
<td>07 Start of Unit-0f-Recovery (UOR)</td>
<td>236</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>12 End of phase 2 syncpoint (DBCTL)</td>
<td>239</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>66 DC enqueue for MSC, 3600 etc</td>
<td>663</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>C5 *** User record ***</td>
<td>63</td>
<td>104</td>
<td>104</td>
</tr>
<tr>
<td>C6 *** User record ***</td>
<td>114</td>
<td>104</td>
<td>104</td>
</tr>
</tbody>
</table>

| Total              | 9063 | 16 | 2907 | 1614725 |

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

**DRL3Inn1 (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 83 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>Num</th>
<th>S</th>
<th>Transact Date and Time</th>
<th>Type</th>
<th>Origin</th>
<th>Dest</th>
<th>Pgm_Name</th>
<th>Region</th>
<th>In,O</th>
<th>Proc</th>
<th>OutQ</th>
<th>Trans Net</th>
<th>CPU</th>
<th>DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>000001</td>
<td>F</td>
<td>00-04-15</td>
<td>07.52.17.5</td>
<td>MTOPRINT</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000002</td>
<td>F</td>
<td>00-04-15</td>
<td>07.52.17.8</td>
<td>MTOPRINT</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000003</td>
<td>F</td>
<td>00-04-15</td>
<td>07.52.31.4</td>
<td>MTOPRINT</td>
<td>0.0</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000004</td>
<td>F</td>
<td>00-04-15</td>
<td>07.52.31.5</td>
<td>BPP9902 IMSMP02</td>
<td>01</td>
<td>0.1</td>
<td>31</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000005</td>
<td>F</td>
<td>00-14-05</td>
<td>07.52.42.2</td>
<td>PGM11000 IMSMP01</td>
<td>01</td>
<td>0.7</td>
<td>13</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000006</td>
<td>F</td>
<td>00-14-05</td>
<td>07.52.42.9</td>
<td>TRN80102 PGM01000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>15.1</td>
<td>0.3</td>
<td>17</td>
<td>3</td>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000007</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.00.2</td>
<td>LTRM0000</td>
<td>TRN80103 PGM01000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.0</td>
<td>0.3</td>
<td>3.7</td>
<td>5</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000008</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.03.9</td>
<td>LTRM0002</td>
<td>TRN81200 PGM21000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.2</td>
<td>1.5</td>
<td>0.0</td>
<td>1.7</td>
<td>2.1</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>000009</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.04.8</td>
<td>LTRM0001</td>
<td>TRN81210 PGM21000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.8</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
<td>1.9</td>
<td>1</td>
</tr>
<tr>
<td>000010</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.09.3</td>
<td>TRN80003 PGM00000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.1</td>
<td>0.1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.1</td>
<td>1.2</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>000011</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.09.1</td>
<td>TRN80002</td>
<td>TRN81220 PGM22000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.9</td>
<td>2</td>
</tr>
<tr>
<td>000012</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.09.5</td>
<td>LTRM0003</td>
<td>TRN81230 PGM22000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.9</td>
<td>1.3</td>
</tr>
<tr>
<td>000013</td>
<td>F</td>
<td>00-14-05</td>
<td>07.53.12.0</td>
<td>TRN80003 PGM00000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.0</td>
<td>0.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>000014</td>
<td>F</td>
<td>00-14-05</td>
<td>07.55.06.4</td>
<td>TRN80005 PGM11000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.1</td>
<td>0.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.3</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>000015</td>
<td>F</td>
<td>00-14-05</td>
<td>07.55.06.8</td>
<td>LTRM0022 PGM11000 IMSMPRT1</td>
<td>06</td>
<td>07</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>000016</td>
<td>F</td>
<td>00-14-05</td>
<td>08.00.10.6</td>
<td>DSFMTCNT DSFMTCNT</td>
<td>09</td>
<td>0.2</td>
<td>30</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000017</td>
<td>F</td>
<td>00-14-05</td>
<td>08.00.10.5</td>
<td>BPP99001 IMSMP99</td>
<td>09</td>
<td>0.2</td>
<td>30</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000018</td>
<td>F</td>
<td>00-14-05</td>
<td>08.28.31.2</td>
<td>IRP99900</td>
<td>BPP99007</td>
<td>09</td>
<td>0</td>
<td>24.1</td>
<td>4255</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 8. Example of R1 report**

**DRL3Inn2 (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 z/OS 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 z/OS 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 z/OS data flow

Figure 9 on page 24 shows an overview of the flow of data within Tivoli Decision Support for z/OS. 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 z/OS.

2. You initiate the collect either through the dialog, or by using a Tivoli Decision Support for z/OS 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:
Understanding 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 z/OS record definition.

4. Tivoli Decision Support for z/OS 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 z/OS record definition.

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

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

9. Tivoli Decision Support for z/OS 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 z/OS selects data from the first table as required by the update definitions and performs required data summarization.
   b. updates other data tables as required by the update definitions.
   (Tivoli Decision Support for z/OS 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 z/OS stores the data from a collect, you can display reports on the data. Tivoli Decision Support for z/OS uses a QMF query to select the data for the report.

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

12. Tivoli Decision Support for z/OS 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 DRLOG points to the data set you want to process.

2. The log collector calls the log procedure (DRL2InnL, where nn is the level of IMS you are using; for example, DRL2I71L for IMS/ESA Version 7 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 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 \( mn \) is the level of IMS you are using; for example, DRL2I71L for IMS/ESA 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. 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
Understanding data flow

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, DRL3Inn3 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 71 or 81. The characters nnn stand for the IMS release and can be 710 or 810. 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 z/OS 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 7 Release 1 is specified:

| IMSVER=71 |

Table 3. Parameter summary for DRL2LOGP

<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>CICSNAME</td>
<td>CICS name for system.</td>
<td>CICS</td>
<td>The CICS name defined for the system.</td>
<td>None</td>
</tr>
<tr>
<td>IMSVER</td>
<td>71</td>
<td>71</td>
<td>IMS version and release number.</td>
<td>None</td>
</tr>
<tr>
<td>OTMATRANCODE</td>
<td>YES, NO</td>
<td>NO</td>
<td>Specifies whether the transaction name should be taken either from OTMA point of view or from IMS point of view.</td>
<td>Low Impact</td>
</tr>
<tr>
<td>REPORTS</td>
<td>R, where a is 0 to 9 and A to Z</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 4. Parameter summary for the log procedure

<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>FASTPATH</td>
<td>ESS FFSYNC MSGQ</td>
<td>If the FASTPATH parameter is not specified, all types of Fast Path transactions are processed. If FASTPATH= is specified, all types of Fast Path transactions denoted by the possible values for this parameter are ignored.</td>
<td>Controls which types of Fast Path transactions are ignored.</td>
<td>Medium</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>MTO_TRAFFIC</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>PSEUDOWFICHK</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</td>
<td>None</td>
<td>Specifies whether the collect restarts if the IMS restart record is found. The collect restarts from 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 z/OS 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. The default is 71, which specifies IMS Version 7 Release 1.

**REPORTS=Ra(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...
Administering the IMS Performance feature

DRL3lim1 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 report 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'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.
Administering the IMS Performance feature

IMSIDCHECK=xxxxxx
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=nnnnn
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=nnnn
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.

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

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 z/OS 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 XFA’, specify RECTYPE=FA. The default is FF.
RESOLUTION=xxxxxxx  
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 DUPKEYERT 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>---</td>
<td>---</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 z/OS 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=xxxxxxx
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 z/OS 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=0M
//COLLECT EXEC PGM=DRLPLC,
  // PARM=('SYSTEM=DB21',
  // 'SYSPREFIX=DRLSYS',
  // 'BPREFIX=DRL',
  // 'SHOWSQL=NO',
  // 'SHOWINPUT=NO')
//STEPLIB DD DISP=SHR, DSN=Tivoli Decision Support for z/OS load library
// DRLPLC
// STEPLIB DD DISP=SHR, DSN=DB2 load library
// DRLLOG DD DISP=SHR, DSN=IMS SLDS/OLDS extract log from IMS
// DRLICHK DD DUMMY -- or previously created checkpoint data set
// DRLICHKO DD DUMMY -- or LRECL=32756 output checkpoint data set
// DRLIOUT DD SYSSOUT=*, DCB=(LRECL=80)
// DRLDUMP DD SYSSOUT=*, 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_V710_COLLECT -- IMS V710
  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 DRL$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"APPLID" = 'IMSPCTL ' ;
SET IMS"CTRL_REGION" = 'PIMSCTL ' ;
COLLECT IMS_Vnnn_COLLECT;
```

Figure 15. Sample data set with log collector parameters
Administering the IMS Performance feature

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 z/OS 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 247 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 241 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 or 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.

DRLIRPTnn—output from report Rnn
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 15 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 z/OS log collector functions and maintains the same interface with the log procedure. DRL2LOGP also calls report programs DLR3innX(where \( nn = \) 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 z/OS CNTL library. Figure 16 shows an example job.

```plaintext
//USERIDA  JOB (ACCO00,001),’DRL2LOGP’,
// NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//*
//DRL2LOGP EXEC PGM=DRL2LOGP
//STEPLIB DD DISP=SHR,DSN= Tivoli Decision Support for z/OS load library
//*
//DRLLOG DD DISP=SHR,DSN=IMS SLDS/OLDS extract log from IMS
//*
//DRLICOMP DD DUMMY -- or LRECL=32756 output composite data set
//DRLICH1 DD DUMMY -- or previously created checkpoint data set
//DRLICH2 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=71 -- the default
REPORTS=R0(00-FE)
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.

Running the IMS Light feature

To transfer the IMS logs from remote locations to those where the unique centralized DB2 database is located, you can use the IMS Light feature to reduce the amount of data that needs to be transferred for central processing. You can produce the the R2 report from the complete IMS log by using the DRL2LOGP as a standalone procedure in the remote location. Then, you can transfer the R2 report to the central location where the final collection processing occurs. The report is
reduced by the 90%. This reduction applies only to the transit time component. To run this process, you need to do some preliminary work as described below.

Setting up Load Library for the IMS Light feature

The Load Modules Library necessary to run the DRL2LOGP stand alone in the remote locations where Tivoli Decision Support for z/OS is not installed must to be built and then redistributed.

To build this Load Lib at the central location where Tivoli Decision Support for z/OS is installed, customize and use the following sample job DRLJIMSL in the Tivoli Decision Support for z/OS CNTL library.

```
//DRLJIMSL JOB (ACCTE), '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 z/OS 1.7.0 */
 /*-------------------------------------------------------------------------*/
 /* FUNCTION: */
 /*-------------------------------------------------------------------------*/
 /* PREPARE LOAD LIBRARY TO RUN STANDALONE THE DRL2LOGP IMS LOG*/
 /*-------------------------------------------------------------------------*/
 /* PROCEDURE: */
 /*-------------------------------------------------------------------------*/
 /* NOTES: */
 /*-------------------------------------------------------------------------*/
 /* BEFORE YOU SUBMIT THE JOB: */
 /* - CHECK ALL DATA SET NAMES IN LOWER CASE (INLIB, OUTLIB DD */
 /* STATEMENTS). */
 /*-------------------------------------------------------------------------*/
 /* - CHECK ALL THE PARMATERS CAREFULLY (DISP, VOL NAMES IN */
 /* INLIB AND OUTLIB DD STATEMENTS). */
 /*-------------------------------------------------------------------------*/
 /* INLIB CONTAINS THE TDS SMP INSTALLED LIBRARY. */
 /* OUTLIB CONTAINS THE OUTPUT COPY LOAD LIBRARY */
 /*-------------------------------------------------------------------------*/
 /* - DECIDE WHETHER SMP DISTRIBUTION ON TARGET LIBRARIES ARE */
 /*-------------------------------------------------------------------------*/
 /* SUITABLE TO BE USED AS SOURCE AND UPDATE INLIB DSN */
 /*-------------------------------------------------------------------------*/
 /* ACCORDINGLY (SDRLLOAD OR ADRLOAD) */
 /*-------------------------------------------------------------------------*/
 /* - DELETE THE SELECT STATEMENT ROW CORRESPONDING TO THE */
 /* IMS RELEASE YOU ARE NOT INTERESTED IN, IF ANY: */
 /*-------------------------------------------------------------------------*/
 /* DRL2171L... FOR IMS V7R1M0 */
 /*-------------------------------------------------------------------------*/
 /* - DOUBLE CHECK IN THE JOB OUTPUT THAT ALL THE NEEDED */
 /* LOAD MODULES HAVE BEEN CORRECTLY COPIED. */
 /*-------------------------------------------------------------------------*/
 /*-------------------------------------------------------------------------*/
 /* STEP1 EXEC PGM=IEBCOPY */
 /*-------------------------------------------------------------------------*/
 /* SYSIN DD SYSOUT=A */
 /*-------------------------------------------------------------------------*/
 /* INLIB DD DISP=(SHR,KEEP),UNIT=3390, VOL=SER=yyyyy, */
 /*-------------------------------------------------------------------------*/
 /* DSN=drl170.SDRLLOAD */
 /*-------------------------------------------------------------------------*/
 /* OUTLIB DD DSN=drl170.ILSMILIGHT.LOAD,DISP=(NEW,CATLG), */
 /*-------------------------------------------------------------------------*/
 /* SPACE=(CYL,(1,1,5)),UNIT=3390, VOL=SER=zzzzzz, */
 /*-------------------------------------------------------------------------*/
 /* DCB=(RECFM=U,BLKSZ=32760,LRECL=0) */
 /*-------------------------------------------------------------------------*/
 /* SYSSIN DD * */
 /*-------------------------------------------------------------------------*/
 /* COPY I=INLIB,O=OUTLIB */
 /*-------------------------------------------------------------------------*/
 /* SELECT MEMBER=(DRL2LOGP, DRL2LOGJ, DRL2IMSL, DRLPM2C, DRLPMSG) */
 /*-------------------------------------------------------------------------*/
 /* SELECT MEMBER=(DRL2I71L, DRL3I710, DRL3I711, DRL3I712) */
 /*-------------------------------------------------------------------------*/
 /*-------------------------------------------------------------------------*/

Statement description

**INLIB** The input dataset points to the Tivoli Decision Support for z/OS SMP/E installation Target or Distribution load library (SDRLLOAD or ADRLOAD) from which the required load modules are copied.
OUTLIB
The output dataset points to the copy output LOADLIB. It will be used in the remote locations as STEPLIB in the DRL2LOGP job.

If you are interested only to a specific version of IMS, the SELECT MEMBER statements related to the other IMS versions (DRL2xxL ... and DRL3xx2) can be deleted from the job. After building the IMSLIGHT.LOAD library, distribute it to the remote centers where it is required. Ensure that the STEPLIB in the DRL2LOGP job used in the required centers is accordingly updated.

//DRLJLOGP JOB (ACCT£),'DRL2LOGP IMS'
//**************************************************************
//*
//*
LICENSED MATERIALS - PROPERTY OF IBM *
//*
//* 5695-101 (C) COPYRIGHT IBM CORPORATION 2003 *
//* SEE COPYRIGHT INSTRUCTIONS. *
//*
//**************************************************************
//*
//* NAME: DRLJLOGP *
//*
//* STATUS: Tivoli Decision Support for z/OS *
//*
//* FUNCTION: *
//* RUN THE IMS LOG PROCEDURE STAND ALONE *
//*
//* NOTES: *
//* BEFORE YOU SUBMIT THE JOB: *
//* - CHECK ALL DATA SET NAMES. *
//* - CHECK ALL THE PARAMETERS CAREFULLY. *
//*
//**************************************************************
//DRL2LOGP EXEC PGM=DRL2LOGP,PARM=('SYSTEM=DSN SYSPREFIX=DRLSYS')
//STEPLIB DD DISP=SHR,DSN=DSN710.IMSLIGHT.LOAD
// DD DISP=SHR,DSN=DSN710.DSNLOAD
//DDLOG DD DISP=SHR,DSN=... INPUT IMS SLEDS LOG
//DRLICOMP DD DISP=SHR,DSN=... OUTPUT COMPOSITE RECORDS OR DUMMY
//DRLICOMK DD DISP=SHR,DSN=... INPUT IMS CHECKPOINT FILE OR DUMMY
//DRLICOMKO DD DISP=SHR,DSN=... OUTPUT IMS CHECKPOINT FILE OR DUMMY
//DRLOUT DD SYSOUT=*,DCB=(RECFM=F,LRECL=80)
//DRLDUMP DD SYSOUT=*,DCB=(RECFM=VB,LRECL=32756)
//DRLIRPT2 DD DISP=SHR,DSN=... OUTPUT RECORDS FROM R2 REPORT/PROCEDURE
//DRLIPARM DD *
IMSID= -- IMS ID FOR THIS IMS SYSTEM
IMSVER=71 -- 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 z/OS 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 COLLECT IMS_V710_COLLECL.
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 z/OS’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 DRL3nnn2 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 z/OS system tables. They define each log to the IMS Performance feature. A log must be defined to Tivoli Decision Support for z/OS before any data can be collected. You specify the log definition that you want to use in the COLLECT statement. 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 z/OS 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.
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:

- **DFSILTA0**: IMS Log Transaction Analysis utility
- **DFSULTA0**: IMS Fast Path Log Analysis utility
- **IMSPARS**: IMS Performance Analysis and Reporting System
- **SLR 3.3**: Service Level Reporter Version 3 Release 3

<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 z/OS/IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'01'</td>
<td>Message received from a CNT</td>
<td>X</td>
<td>X</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>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'07'</td>
<td>Program termination</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'08'</td>
<td>Program initiation</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'11'</td>
<td>Start of conversation</td>
<td></td>
<td>X</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>X</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>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'33'</td>
<td>Message queue free</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'34'</td>
<td>Message cancel</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'35'</td>
<td>Message queue enqueue</td>
<td>X</td>
<td>X</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>X</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>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>X'4098'</td>
<td>IMS checkpoint end</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</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></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
Log and record definitions

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 z/OS/IMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>X'5936'</td>
<td>Fast Path dequeue</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5937'</td>
<td>Fast Path syncpoint</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>X'5938'</td>
<td>Fast Path abend</td>
<td></td>
<td>X</td>
<td></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 V710 signifies IMS Version 7 Release 1)

**record type** Corresponding IMS record type

*Table 6 on page 49* 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). This record represents the message and its text and control information as it appears on the IMS message queues. The DRRN indicates the message queue type: X’00......’ indicates QBLKS X’04......’ indicates short message queue X’08......’ indicates long message queue The record contains the indicator of the origin of the message, its destination, whether it uses MSC, and so on. Note that IMS writes this record when a terminal or another network attached system receives a message from a CNT (such as MSC/ISC and FES). It may be input to a program (if it is enqueued to an SMB) or it may switch to another CNT (the latter is known as a message switch). The only occasion when the date and time fields represent an approximation of when the event occurred is for the originating message. Be careful when using these fields for program-to-program switches and MSC/ISC/FES activity.</td>
</tr>
</tbody>
</table>
| X'02'       | IMS_Vnnn_02       | Command log record. This record represents a condensed version of the command entered by the IMS user or program. If the command string requires a record longer than the logical record length of the log, the record is segmented into several type X'02' records.
### 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 the number of DL/1 calls cannot be calculated for each message or commit. The only way that the amount of dependent region CPU and number of DL/1 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>
</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'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 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>
<tr>
<td>Record type</td>
<td>Record definition</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| X'30'       | IMS_Vnnn_30       | Message queue prefix changed record.  
This record logs changes made to the message queue record prefix. It links directly to a previously logged message. |
| X'31'       | IMS_Vnnn_31       | Message queue GU record.  
This record logs the details of a message that is GU’d from the message queue to be sent to its destination. The destination may be an SMB or CNT.  
The record is present for incoming messages that are processed by a program scheduled in a message processing region, or for outgoing messages that are sent to a network destination. In addition, the record is present for message switches.  
The timestamp in this record essentially represents the time that the message ceased waiting on the message queue. If the message is sent to a CNT, a type X'36' record follows, ultimately being followed by a type X'33' Free record, regardless of destination. |
| X'32'       | IMS_Vnnn_32       | Message queue reject record.  
This record is produced when the MSGQ rejects a message because an error occurred, presumably causing an application program abend. |
| X'33'       | IMS_Vnnn_33       | Message queue DRRN free record.  
This record indicates that DRRNs were freed from the message queues, the message was deleted, and the DRRNs are available for reuse.  
This record always indicates that the message is no longer needed by IMS. |
| X'34'       | IMS_Vnnn_34       | Message queue cancel record.  
This record indicates that the message was canceled from the queue and that a subsequent X'35' Enqueue record was not produced. |
| X'35'       | IMS_Vnnn_35       | Message queue enqueue record.  
This record indicates that the message in the message queue (logged as a type X'01' or X'03' record) has been placed on the queue for processing. If the destination is an SMB, it is usually waiting on the input queue for the PSB to issue the Get Unique. If the destination is a CNT, it is either a message switch or an outbound message and is waiting in the output queue.  
This record follows the logging of the message as a X'01' or X'03' record, and precedes the X'31' record that indicates the message has been retrieved from the queue for processing or transmission.  
The timestamp can be carried from the preceding X'01' or X'03'. Because the timestamp from this record may not reflect the real time the message arrived in the system, exercise caution when using it. |
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>
### Log and record definitions

<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>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 synpoint record. This record indicates that a successful Fast Path synpoint occurred, indicating that any messages can be transmitted.</td>
</tr>
<tr>
<td>X'5938'</td>
<td>IMS_Vnnn_5938</td>
<td>EMH Fast Path synpoint failure record. This record indicates that a Fast Path synpoint 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>
</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’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></td>
<td>Buffered Fast Path.</td>
</tr>
<tr>
<td>X’5E’</td>
<td></td>
<td>Image capture of SB handler.</td>
</tr>
<tr>
<td>X’5F’</td>
<td></td>
<td>DL/I call trace.</td>
</tr>
<tr>
<td>X’62’</td>
<td></td>
<td>OSAM error.</td>
</tr>
<tr>
<td>X’63’</td>
<td></td>
<td>Allocate/deallocate.</td>
</tr>
<tr>
<td>X’64’</td>
<td></td>
<td>Message discarded by MSC.</td>
</tr>
<tr>
<td>X’65’</td>
<td></td>
<td>IRSS and SNA restart.</td>
</tr>
<tr>
<td>X’66’</td>
<td></td>
<td>3600 standard record.</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></td>
<td>3275 switched unauthorized ID.</td>
</tr>
<tr>
<td>X’6C’</td>
<td></td>
<td>MSC link connect/disconnect.</td>
</tr>
<tr>
<td>X’6D’</td>
<td></td>
<td>XRF hot standby surveillance.</td>
</tr>
<tr>
<td>X’6E’</td>
<td></td>
<td>XRF session miscellaneous.</td>
</tr>
<tr>
<td>X’70’</td>
<td></td>
<td>Online change.</td>
</tr>
<tr>
<td>X’71’</td>
<td></td>
<td>TCF record.</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'xxFF', X'xxFE', X'xxFD', X'xxFC', X'xxFB', X'xxFA', X'xxEF', X'xxEE', X'xxED', X'xxEC', 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 62](#) explains the sections of the composite record:

<table>
<thead>
<tr>
<th>Short description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The actual name of the section</td>
</tr>
<tr>
<td>Prefix</td>
<td>The one-character prefix for all data subsections and fields defined in the record section</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>
</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>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>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 X'FA', 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>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><strong>Message queue output</strong></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.</td>
</tr>
<tr>
<td><strong>Note:</strong> If the composite record type and subtype is XFB, 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. &lt;br&gt; 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>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Program</strong></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. &lt;br&gt; <strong>Note:</strong> 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. &lt;br&gt; 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 64 for more information about X'5937' and X'5938' records.) &lt;br&gt; 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. &lt;br&gt; 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>

---

Log and record definitions
### Log and record definitions

**Table 7. Composite record sections in IMS_Vn.nn_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</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>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 65), 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</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>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 65), 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>
</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>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:

- 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>
<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>-</td>
<td>X</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>X</td>
<td>-</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 z/OS database is a collection of DB2 tables. Each table contains a fixed number of columns. The number of rows in each table varies with time, because of rows added by the collect function and because of database maintenance.

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

This 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:

IMS_content_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:

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>The table holds nonsummarized data (timestamped data).</td>
</tr>
<tr>
<td>Q</td>
<td>The table holds data summarized by quarter hour.</td>
</tr>
<tr>
<td>H</td>
<td>The table holds data summarized by hour.</td>
</tr>
<tr>
<td>D</td>
<td>The table holds data summarized by day.</td>
</tr>
<tr>
<td>W</td>
<td>The table holds data summarized by week.</td>
</tr>
<tr>
<td>M</td>
<td>The table holds data summarized by month.</td>
</tr>
</tbody>
</table>

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 71 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 z/OS 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 z/OS 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
• **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
• Supported by record definitions IMS_Vnnn_COMP and IMS_Vnnn_R2 with record procedure definition DRL3I

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 71***
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>
<thead>
<tr>
<th>Table:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_APPLICATION_W</td>
<td></td>
</tr>
<tr>
<td>IMS_APPLICATION_H</td>
<td></td>
</tr>
<tr>
<td>IMS_USER_APPL_W</td>
<td></td>
</tr>
<tr>
<td>IMS_USER_APPL_D</td>
<td></td>
</tr>
<tr>
<td>IMS_SYSTEM_D</td>
<td></td>
</tr>
<tr>
<td>IMS_SYSTEM_O</td>
<td></td>
</tr>
<tr>
<td>IMS_TRANSACTION_W</td>
<td></td>
</tr>
<tr>
<td>IMS_TRANSACTION_H</td>
<td></td>
</tr>
<tr>
<td>IMS_USER_TRAN_W</td>
<td></td>
</tr>
<tr>
<td>IMS_USER_TRAN_D</td>
<td></td>
</tr>
<tr>
<td>IMS_USER_TRAN_H</td>
<td></td>
</tr>
</tbody>
</table>

**Key columns:**

| Column                        | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS | IMS |
|-------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| NPS_SYSTEM_ID                 | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   |
| IMS_SYSTEM_ID                 |     | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   |     |     |     |     |     |     |     |     |     |     |
| APPLICATION_NAME              |     |     |     |     |     |     |     |     |     |     | X   | X   |     |     |     |     |     |     |     |     |     |     |
| TRANSACTION_NAME              | X   | X   | X   | X   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| TRANSACTION_CLASS             |     | X   | X   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| PROGRAM_NAME                  |     |     |     | X   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| IMS_TRAN_TYPE                 |     |     |     |     |     |     |     |     |     |     |     |     |     |     | X   |     |     |     |     |     |     |     |
| REGION_JOB_NAME               |     |     |     |     | X   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| REGION_PST_ID                |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     | X   |     |     |     |     |     |
| USER_ID                       |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     | X   |     |     |
| LOGICAL_TERMINAL             |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     | X   |
| PHYSICAL_TERMINAL            |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| PERIOD_NAME                   | X   | X   | X   | X   | X   |     |     |     | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   | X   |     |     |     |
| DATE                          |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| TIME                          | X   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |

*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:
- IMS_TRANSACTION_H: 1 day
- IMS_TRANSACTION_D: 35 days
- IMS_TRANSACTION_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>k</td>
<td>DATE</td>
</tr>
<tr>
<td>TIME</td>
<td>k</td>
<td>TIME</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td>k</td>
<td>CHAR(2)</td>
</tr>
</tbody>
</table>

The date the activities occurred. For the _W table, this is the date of the first day of the week.

The time when the activity started, in the format HH.00.00. This applies only to the _H table.

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.

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.

The name of the IMS transaction the user requested.

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.

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.

The assigned transaction class.
## 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>k CHAR(2)</td>
<td>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)</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td>k CHAR(2)</td>
<td>The assigned transaction class.</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>r CHAR(8)</td>
<td>The VTAM applid for the IMS system. 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_CTRL_REGION</td>
<td>r 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>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>APPC_TPI_COMMITS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>APPC_TPI_CPU_SEC</td>
<td>FLOAT</td>
<td>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).</td>
</tr>
<tr>
<td>APPC_TPI_DPSBCALLS</td>
<td>FLOAT</td>
<td>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.</td>
</tr>
<tr>
<td>APPC_TPI_ENDED</td>
<td>FLOAT</td>
<td>The total number of APPC TPI termination records (record type X'0A07'). This represents the number of APPC transaction program instance terminations.</td>
</tr>
<tr>
<td>APPC_TPI_STARTED</td>
<td>FLOAT</td>
<td>The total number of APPC TPI schedule records (record type X'0A08'). This represents the number of APPC transaction program instance schedules.</td>
</tr>
<tr>
<td>APPC_NETID</td>
<td>CHAR(8)</td>
<td>The destination network ID for the APPC 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>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>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>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>DLI_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>DLIDB_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>DLIDBCTL_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>
<tr>
<td>DL1_EXCL_DEQUEUES</td>
<td>FLOAT</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'). 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.</td>
</tr>
<tr>
<td>DL1_EXCL_ENQUEUES</td>
<td>FLOAT</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'). 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.</td>
</tr>
<tr>
<td>DL1_EXCL_ENQWAITS</td>
<td>FLOAT</td>
<td>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.</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>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_DEQUEUE</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_ENQUEUE</td>
<td>FLOAT</td>
<td>The total number 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_ENQUEUE</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_DEQUEUE</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_ENQUEUE</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_ENQUEUE</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_UPDT_DEQUEUE</td>
<td>FLOAT</td>
<td>The total number of DL/I update dequeue calls, 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_ENQUEUE</td>
<td>FLOAT</td>
<td>The total number of DL/I update enqueue calls, 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_ENQUEUE</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>
</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_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 DEDB calls by HSSP 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_NNH_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_SMMSG_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_SMMSG_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>
</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>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_RESPONSES</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>
</tbody>
</table>
| PGM_TIMER_UNITS       | FLOAT     | 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.         |
| RESPONSE_BNDY1_SEC    | FLOAT     | The first transaction transit-time boundary, in seconds (default is 1 second). This is used to determine the number of transactions within this boundary. |
| RESPONSE_BNDY2_SEC    | FLOAT     | The second transaction transit-time boundary, in seconds (default is 2 seconds). This is used to determine the number of transactions within this boundary. |
| RESPONSE_BNDY3_SEC    | FLOAT     | The third transaction transit-time boundary, in seconds (default is 5 seconds). This is used to determine the number of transactions within this boundary. |
| RESPONSE_BNDY4_SEC    | FLOAT     | The fourth transaction transit-time boundary, in seconds (default is 10 seconds). This is used to determine the number of transactions within this boundary. |
Data tables and lookup tables

**IMS_USER_TRAN_H, _D, _W**

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 71

<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</td>
</tr>
<tr>
<td>TIME</td>
<td>k</td>
<td>TIME</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>USER_ID</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>k</td>
<td>CHAR(8)</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_TRAN_TYPE</td>
<td>k</td>
<td>The transaction characteristics, including transaction type, region type, program-to-program switch flag, MSC/ISC flag, mixed mode flag, completion status flag, and abend status flag. This column applies only to the _H table. For more information about this column, see <a href="#">IMS_TRAN_TYPE key column</a></td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k</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>

**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, _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:
IMS_SYSTEM_Q 70 days
IMS_SYSTEM_D 1095 days

<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 activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k 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>k 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>k CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2I.nnL 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>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>REGION_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>TRANSACTION_CLASS</td>
<td>k 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>k</td>
<td>CHAR(2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The transaction type based on (DRL_PRTM + DRL_IQTM) / 10 value ranges. Possible values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F = FAST (0.0 - 0.8)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>G = GOOD (0.8 - 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>
</tbody>
</table>

| 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)                                                         |
|                   |           | S = SLOW (above 10)                                                          |

| PERIOD_NAME       | k         | CHAR(8)                                                                      |
|                   |           | The name of the period or shift in which the activity occurred, for example, PRIME shift 08:00 to 17:00 weekdays. This column is derived using the MVS_SYSTEM_ID, DATE, and TIME columns as parameters in the PERIOD function. |

**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 71.
Application subcomponent tables

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 IMS_APPLICATION 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 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>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 71.
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 DRL2InnnL 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 /SIGNON 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 71.
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_STxxx.

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 DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k 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>k CHAR(4)</td>
<td>The MVS (SMF) system ID defined in SYS1.PARMLIB(SMFFPRMnn) 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 CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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>BUFFER_SIZE</td>
<td>k CHAR(8)</td>
<td>The size of buffers in this buffer pool.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k 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>r 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>r SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r 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>
</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.AlterCalls</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 idents 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.RLSEOWN</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>k DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k 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>k 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>k CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID.</td>
</tr>
<tr>
<td>POOL_ID</td>
<td>k CHAR(8)</td>
<td>The name of the buffer pool for the gathered statistics.</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k 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>r 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>r SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r 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>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>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>
Data tables and lookup tables

**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>k DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH.MM.SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k 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>k 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>k CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2ImL 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>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>REGION_PST_ID</td>
<td>k 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>k 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>r 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>r SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r 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>
</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</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>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>k DATE</td>
<td>The date the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k TIME</td>
<td>The time when the activity started, in the format HH:MM:SS.</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k 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>k 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>k CHAR(8)</td>
<td>The IMS subsystem ID defined in the IMS log procedure parameter IMSID. The log procedure DRL2lnnL 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>PERIOD_NAME</td>
<td>k 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>r 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>r SMALLINT</td>
<td>The ascending numeric ID of the checkpoint for the IMS session.</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r 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>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>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_WTBFCHKPT</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_WTT_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_WTTWT_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_SYNONYMMS_MAX</td>
<td>INTEGER</td>
<td>The maximum number of program isolation synonym searches.</td>
</tr>
<tr>
<td>PI_SYNONYMMS_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_WRITEWAIT</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_BFRWAIT</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_PCI_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_LOCATORS</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>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<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>
# Data tables and lookup tables

## 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>k</td>
<td>DATE</td>
</tr>
<tr>
<td>TIME</td>
<td>k</td>
<td>TIME</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>k</td>
<td>TIMESTAMP</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>BUFFER_SIZE</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>IMS_APPLID</td>
<td>r</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>IMS_CHECKPOINT</td>
<td>r</td>
<td>SMALLINT</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>r</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>r</td>
<td>CHAR(4)</td>
</tr>
<tr>
<td>BACKGRND_WRITE_REQ</td>
<td>FLOAT</td>
<td>The number of background write requests.</td>
</tr>
<tr>
<td>BUFFER_ERRORS</td>
<td>FLOAT</td>
<td>The number of buffers currently write-error marked for the IMS session.</td>
</tr>
<tr>
<td>BUFFER_ERRORS_MAX</td>
<td>FLOAT</td>
<td>The maximum number of buffers write-error marked for the IMS session.</td>
</tr>
<tr>
<td>BUFFERS</td>
<td>FLOAT</td>
<td>The number of buffers in the buffer pool for the IMS session.</td>
</tr>
<tr>
<td>CI_FOUND_IN_POOL</td>
<td>FLOAT</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>FLOAT</td>
<td>The number of hiperspace buffers for the subpool.</td>
</tr>
<tr>
<td>HS_READS</td>
<td>FLOAT</td>
<td>The number of successful VSAM reads from hiperspace.</td>
</tr>
<tr>
<td>HS_READS_FAILED</td>
<td>FLOAT</td>
<td>The number of failed VSAM reads from hiperspace.</td>
</tr>
<tr>
<td>HS_WRITES</td>
<td>FLOAT</td>
<td>The number of successful VSAM writes to hiperspace.</td>
</tr>
<tr>
<td>HS_WRITES_FAILED</td>
<td>FLOAT</td>
<td>The number of failed VSAM writes to hiperspace.</td>
</tr>
<tr>
<td>LRECL_ALTERED</td>
<td>FLOAT</td>
<td>The number of logical records marked altered for the IMS session.</td>
</tr>
<tr>
<td>PLH_Waits</td>
<td>FLOAT</td>
<td>The number of place holder 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>FLOAT</td>
<td>The number of requests to retrieve by key for the IMS session.</td>
</tr>
<tr>
<td>RETRIEVES_BY_RBA</td>
<td>FLOAT</td>
<td>The number of requests to retrieve by RBA for the IMS session.</td>
</tr>
<tr>
<td>SYNC_CALLS</td>
<td>FLOAT</td>
<td>The number of synchronization calls for the IMS session.</td>
</tr>
<tr>
<td>VSAM_ESDS_INSERTS</td>
<td>FLOAT</td>
<td>The number of logical records inserted to ESDS for the IMS session.</td>
</tr>
<tr>
<td>VSAM_GET_CALLS</td>
<td>FLOAT</td>
<td>The number of VSAM get calls issued for the IMS session.</td>
</tr>
<tr>
<td>VSAM_KSDS_INSERTS</td>
<td>FLOAT</td>
<td>The number of logical records inserted to KSDS for the IMS session.</td>
</tr>
<tr>
<td>VSAM_NUSER_WRITES</td>
<td>FLOAT</td>
<td>The number of VSAM space write requests for the IMS session.</td>
</tr>
<tr>
<td>VSAM_READS</td>
<td>FLOAT</td>
<td>The number of VSAM read I/O operations for the IMS session.</td>
</tr>
<tr>
<td>VSAM_SCHBFR_CALLS</td>
<td>FLOAT</td>
<td>The number of VSAM SCHBFR calls issued for the IMS session.</td>
</tr>
<tr>
<td>VSAM_USER_WRITES</td>
<td>FLOAT</td>
<td>The number of VSAM user write requests for the IMS session.</td>
</tr>
</tbody>
</table>
Data tables and lookup tables

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 CHAR(4)</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 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>TRANSACTION_NAME</td>
<td>k CHAR(8)</td>
<td>The name of the IMS transaction.</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>APPLICATION_NAME</td>
<td>CHAR(18)</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</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td>k</td>
<td>CHAR(8)</td>
</tr>
<tr>
<td>GROUP_ID</td>
<td></td>
<td>CHAR(8)</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

<table>
<thead>
<tr>
<th>Chapter 7. Introduction to the IMS Shared Queue feature</th>
<th>Chapter 11. IMS Shared Queue record definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding the IMS Shared Queue feature</td>
<td>Composite record definitions</td>
</tr>
<tr>
<td>Collecting data</td>
<td>Composite record sections</td>
</tr>
<tr>
<td>SLDS</td>
<td></td>
</tr>
<tr>
<td>Log procedure</td>
<td></td>
</tr>
<tr>
<td>Composite record</td>
<td></td>
</tr>
<tr>
<td>Record procedure</td>
<td></td>
</tr>
<tr>
<td>Record definitions</td>
<td></td>
</tr>
<tr>
<td>Tivoli Decision Support for z/OS data tables and</td>
<td></td>
</tr>
<tr>
<td>environmental information</td>
<td></td>
</tr>
<tr>
<td>Reports</td>
<td></td>
</tr>
<tr>
<td>The log collector and DRLSLOGP</td>
<td></td>
</tr>
<tr>
<td>Installing and customizing the IMS Shared Queue feature</td>
<td></td>
</tr>
<tr>
<td>Planning for the IMS Shared Queue feature</td>
<td></td>
</tr>
<tr>
<td>Selecting IMS Shared Queue feature components</td>
<td></td>
</tr>
<tr>
<td>The collect components</td>
<td></td>
</tr>
<tr>
<td>The log records component</td>
<td></td>
</tr>
<tr>
<td>Updating other lookup and control tables</td>
<td></td>
</tr>
<tr>
<td>Using the IMS Shared Queue feature</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8. Using log and record procedures within the IMS Shared Queue</th>
<th>Chapter 12. IMS Shared Queue data tables and lookup tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>The log procedure</td>
<td>IMS Shared Queue data tables</td>
</tr>
<tr>
<td>Set relationships</td>
<td>Transaction Transit Time subcomponent tables and views</td>
</tr>
<tr>
<td>Composite records and subtypes</td>
<td>IMS TRAN_QUEUE_QV_D</td>
</tr>
<tr>
<td>Handling of special IMS cases within the IMS Shared Queue</td>
<td>IMS SYSTEM_TRAN_H_D</td>
</tr>
<tr>
<td>Log procedure DRLOUT reports</td>
<td>TRANS_TYPE key column</td>
</tr>
<tr>
<td></td>
<td>Views</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_QV_D W</td>
</tr>
<tr>
<td></td>
<td>Account and Availability subcomponent tables</td>
</tr>
<tr>
<td></td>
<td>Data Tables</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_QV_D DV</td>
</tr>
<tr>
<td></td>
<td>IMS SYSTEM_TRAN_H_D</td>
</tr>
<tr>
<td></td>
<td>TRANS_TYPE key column</td>
</tr>
<tr>
<td></td>
<td>Views</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_QV_D W</td>
</tr>
<tr>
<td></td>
<td>Data Tables</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_QV_D DV</td>
</tr>
<tr>
<td></td>
<td>Lookup tables</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_QV_D W</td>
</tr>
<tr>
<td></td>
<td>Lookup tables</td>
</tr>
<tr>
<td></td>
<td>Example of Table Contents</td>
</tr>
<tr>
<td></td>
<td>Mapping between Non-SQ and SQ DB2 Tables</td>
</tr>
<tr>
<td></td>
<td>IMS TRANSACTION_H, D, W</td>
</tr>
<tr>
<td></td>
<td>IMS USER_TRAN_x (H, D, W)</td>
</tr>
<tr>
<td></td>
<td>IMS SYSTEM_x (Q, D)</td>
</tr>
<tr>
<td></td>
<td>IMS APPLICATION_x (H, W)</td>
</tr>
<tr>
<td></td>
<td>IMS USER_APPL_x (D, W)</td>
</tr>
<tr>
<td></td>
<td>Mapping between New DB2 Table Fields and CSQ Records</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_x (H, D, W)</td>
</tr>
<tr>
<td></td>
<td>IMS TRAN_QUEUE_x (Q, D)</td>
</tr>
<tr>
<td></td>
<td>IMS SYSTEM_TRAN_x (H, D)</td>
</tr>
<tr>
<td></td>
<td>IMS PSB_ACCOUNT_x (H, D, W)</td>
</tr>
<tr>
<td></td>
<td>New IMS CSQ Composite Record Header Layout (DRLCSQCR)</td>
</tr>
<tr>
<td></td>
<td>Mapping between SQ and non-SQ IMS R2 Fields</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 9. Understanding data flow through IMS Shared Queue</th>
<th>Chapter 10. Administering the IMS Shared Queue feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log collector data flow</td>
<td>IMS Shared Queue Logs</td>
</tr>
<tr>
<td>DRLSLOGP data flow</td>
<td>Specifying DRLSLOGP and log procedure parameters</td>
</tr>
<tr>
<td>IMS Shared Queue Collect feature data flow</td>
<td>Specifying the log collector parameter</td>
</tr>
<tr>
<td></td>
<td>Running the log collector in a shared queue environment</td>
</tr>
<tr>
<td></td>
<td>Using DRLSLOGP</td>
</tr>
<tr>
<td></td>
<td>DRLSLOGP input and output data sets</td>
</tr>
<tr>
<td></td>
<td>Running DRLSLOGP</td>
</tr>
<tr>
<td></td>
<td>Operational considerations</td>
</tr>
<tr>
<td></td>
<td>Running the IMS Light feature</td>
</tr>
<tr>
<td></td>
<td>Setting up Load Library for the IMS Light feature</td>
</tr>
<tr>
<td></td>
<td>Statement description</td>
</tr>
<tr>
<td></td>
<td>Setting up the Tivoli Decision Support collect</td>
</tr>
<tr>
<td></td>
<td>for the IMS Light feature</td>
</tr>
</tbody>
</table>

103
Chapter 7. Introduction to the IMS Shared Queue feature

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

The Tivoli Decision Support for z/OS base includes:

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

Each feature provides:

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

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

When you use IMS 7.1 in a non-shared Queue environment, you can decide whether to use the non-Shared Queue support engine (not including the latest enhancements/re-design implementation) or the new Tivoli Decision Support for z/OS 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. To collect data from IMS 8.1, you must use the new engine with the CSQ_V810_COLLECT process. This means that the non-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.

If you migrate from the non-Shared Queue IMS feature to the IMS Shared Queue, you must use the new set of reports. If you want to continue to use your old reports, you must modify them to match the new table and field organization before you start using the IMS Shared Queue feature. This is required because there is not an exact mapping between the non-Shared Queue and Shared Queue fields.

For details about how the DB2 tables and their fields have been re-organized see “Mapping between Non-SQ and SQ DB2 Tables” on page 161.
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 z/OS DB2 tables. [Figure 18] 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.

Log procedure
A Tivoli Decision Support for z/OS 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 z/OS 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 z/OS record procedure. The record definitions are used by Tivoli Decision Support for z/OS collect when updating DB2 tables.

Tivoli Decision Support for z/OS data tables and environmental information
The IMS Shared Queue feature uses the R2 records, along with user-supplied data in Tivoli Decision Support for z/OS 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 z/OS 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 106 you run the Tivoli Decision Support for z/OS log collector program. The log collector uses record definitions and other definitions when it updates Tivoli Decision Support for z/OS data tables. The log collector is part of the Tivoli Decision Support for z/OS 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.
Introduction to the IMS Shared Queue Feature

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.

For data flow diagrams for the log collector and DRLSLOGP, see Chapter 9, “Understanding data flow through IMS Shared Queue,” on page 119. For more information about running the log collector and DRLSLOGP, see Chapter 10, “Administering the IMS Shared Queue feature,” on page 125.

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 z/OS 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 z/OS. The IMS Performance feature supports data from systems running IMS/ESA on Version 7 Release 1 and 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 z/OS 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 z/OS 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 z/OS 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 z/OS collects unnecessary data, which takes up disk space.

The IMS Shared Queue feature components and subcomponents contain Tivoli Decision Support for z/OS 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 z/OS features and components using the procedure in Administration Guide. After the system programmer has successfully installed the Tivoli Decision Support for z/OS base, you can choose whether to install the IMS feature and its components and subcomponents. Tivoli Decision Support for z/OS stores the necessary log, record, and update definitions
Introduction to the IMS Shared Queue Feature

in Tivoli Decision Support for z/OS system tables. Tivoli Decision Support for z/OS also loads predefined DB2 tables and reports.

The IMS Shared Queue feature components are:

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

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

**Transaction Transit Time 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.

**The log records component**
Use the log records component for your own IMS analysis. When you install a log records component, you get access to the Tivoli Decision Support for z/OS record definitions for the IMS records in the SLDS. You can write your own Tivoli Decision Support for z/OS definitions to process IMS records, and, for example, define your own DB2 tables and reports. The log records components do not have any subcomponents, hence they do not update DB2 tables and do not produce reports.

For a description of the Tivoli Decision Support for z/OS record definitions for the IMS records, see Chapter 11, “IMS Shared Queue record definitions,” on page 137.

---

**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 z/OS 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 z/OS 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 1.
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 z/OS 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 z/OS. For specific information about running the IMS Shared Queue feature, see “Administering the IMS Shared Queue feature,” on page 125.
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 origin 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 l0f 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 DRLSimm2 (where $nn$ is 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 z/OS 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
Using lof and record procedures within the IMS Shared Queue

receives the X'33' (GET) record for the input PUOW, it writes the input-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_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
Using log and record procedures within the IMS Shared Queue

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 z/OS 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 [147].

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 bypass 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 118] 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.
Using log and record procedures within the IMS Shared Queue

DRL2070I Batch Driver Parameters:

DRLIPARM Reports requested are RZ(FF)
DRLIPARM IMS version: 71

DRL2071I Parameters used in this run:

Default TABLE FLUSH = 0
Default MAX FREE POINTERS = 5000
Default START = 0000000F 0000000F
Default STOP = 2050365F 2359599F
DRLIPARM IMSID = IMS
Default RECTYPE = FF
DRLIPARM WRITEPENDING = YES
Default RESTARTCHECK = YES
Default STATISTIC = YES
Default ACCOUNT = YES

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.

DRL2072I Statistics for NODEs created this run:

<table>
<thead>
<tr>
<th>NODE type</th>
<th>NODE Initially allocated length</th>
<th>Total NODEs allocated length</th>
<th>NODEs pending</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUOW</td>
<td>40 140000</td>
<td>140000</td>
<td>56</td>
</tr>
<tr>
<td>PUOW</td>
<td>40 380000</td>
<td>380000</td>
<td>101</td>
</tr>
<tr>
<td>PSB</td>
<td>40 180000</td>
<td>180000</td>
<td>71</td>
</tr>
<tr>
<td>IMS</td>
<td>70 5775000</td>
<td>5775000</td>
<td>424</td>
</tr>
<tr>
<td>TOTALS</td>
<td>190 6475000</td>
<td>6475000</td>
<td>652</td>
</tr>
</tbody>
</table>

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.
Chapter 9. Understanding data flow through IMS Shared Queue

This chapter describes the flow of data through the IMS Shared Queue Collect feature. The chapter includes:

• Tivoli Decision Support for z/OS 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 z/OS log collector
  – When generating output composite records using the DRLSLOGP batch program
• IMS Shared Queue Collect feature data flow through the collect component
Understanding data flow through IMS Shared Queue

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 129.

Figure 22. IMS Shared Queue data flow through the log collector

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

1. 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 DRLOG points to the data set you want to process.

2. The log collector calls the log procedure (DRLSINnL, where nn is the level of IMS you are using; for example, DRLS171L for IMS Version 7 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.

**DRLSLOGP data flow**

Figure 23 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 130.

The characters nn indicate the IMS release and can be 71 or 81.

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 (DRLSI\(nn\)L, 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.
Understanding data flow through IMS Shared Queue

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

Figure 24 and Figure 25 on page 123 show the flow of data through the IMS Shared Queue Collect feature when you use the log collector. They show the data flow from the original IMS log data to the IMS Shared Queue Collect feature reports. They give a more inclusive picture of the IMS Shared Queue Collect feature data flow; compare them with Figure 10 on page 26.

The characters nn stand for the IMS release and can be 71 or 81 and the characters nnn stand for the IMS release and can be 710 or 810.
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 z/OS 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 z/OS 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 z/OS 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 z/OS 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 z/OS collect is run. The maximum number of logs merged by the IMS utility is nine. See Appendix E, “DFSLTMG0 log merge utility,” on page 249 for details.

2. To use the online merging option of Tivoli Decision Support for z/OS using the SQNLOGS parameter. This means that in input Tivoli Decision Support for z/OS 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 129 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 7 Release 1 is specified as

IMSVER=71
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>71, 81</td>
<td>71</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 queuing 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.</td>
<td>Medium - High (according to the number of logs and record sequences)</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</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</td>
<td>YES</td>
<td>Controls whether X’40bx7’ records will be written in DRLICOMP dataset as CSQ_Vnn_STxx records.</td>
<td>Low</td>
</tr>
</tbody>
</table>

The parameters for DRLSLOGP are:
Administering the IMS Shared Queue feature

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 71, which specifies IMS
   Version 7 Release 1.

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 DRLSImn2 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. xxxxxxxxx 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 ‘XFA’,
   specify RECTYPE=FA. The default is FF.

RESTARTCHECK=NO
   If you specify No, when an IMS restart record is found, Tivoli Decision
   Support for z/OS 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.
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 ddbname 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 the COUPLExx parmlib 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.

```batch
//USERIDA JOB (ACC000,001),'IMS1 COLLECT',
//NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//COLLECT EXEC PGM=DRLPLC,
//PARM=('SYSTEM=DB21',
//'SYSPREFIX=DRLSYS',
//'&PREFIX=DRL',
//'SHOWSQL=NO',
//'SHOWINPUT=NO')
//STEPLIB DD DISP=SHR,DSN= Tivoli Decision Support for z/OS load library
// DD DISP=SHR,DSN=DB2 load library
//DRLLOG DD DISP=SHR,DSN=(IMS SLDS or IMSes shared merged log or DRLICOMP)
// DRLICHKI DD DUMMY -- or previously created checkpoint data set
//DRLICHKO DD DUMMY -- or LRECL=32756 output checkpoint data set
/*
//DRLOUT DD SYSOUT=**,DCB=(LRECL=80)
//DRLDUMP DD SYSOUT=**,DCB=(LRECL=32756)
//DRLIPARM DD *
ACCOUNT=YES
MAXFREE=3000
/*
//DRLIN DD DISP=SHR,DSN=USERIDA.IMS.DEFS.V61(DRL$CVAR)
// DD *
COLLECT CSQ_V710_COLLECT -- IMS V7
BUF SIZ 50 M; -- or CSQ_V810_COLLECT for IMS V8
/*
```

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=.....
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 DRLSLOGP

DRLSLOGP 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 z/OS environment. DRLSLOGP is the equivalent of running the log procedure under the log collector, except that DB2 is never invoked and DB2 tables are not updated.

DRLSLOGP can create a large amount of output. Anyway the DRLSLOGP 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 z/OS and after send the produced output to main center.

DRLSLOGP input and output data sets

DRLSLOGP 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 z/OS IMS feature. Either dummy this ddname or allocate it to a data set with record length of 32,768 bytes.

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 data set with record length of 32 756 bytes, 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.

**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 feature writes report output to this ddname. You can allocate this ddname to SYSOUT, a physical data set, or dummy.

**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 z/OS 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:
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 (SLU P) 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.

```
//USERIDA JOB (ACC000,001),'DRLSLOGP',
// NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
/*
//DRLLOG1 EXEC PGM=DRLLOG2
//DD DISP=SHR,DSN=Tivoli Decision Support for z/OS load library
//DD DISP=SHR,DSN=dr170.SDRLLOAD
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS1
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS2
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS3
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS4
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS5
//DD DUMMY -- or LRECL=32756 output composite data set
//DD DUMMY -- or LRECL=32756 output checkpoint data set
/*
//DRLCOMP DD DUMMY -- or LRECL=32756 output composite data set
//DRLICOMP DD DUMMY -- or LRECL=32756 output checkpoint data set
/*
//DRLOUT DD SYSOUT**,DCB=(RECFM=F,LRECL=80)
//DRLDUMP DD SYSOUT**,DCB=(RECFM=VB,LRECL=32756)
//DRLRPT2 DD SYSOUT**,DCB=(RECFM=FB,LRECL=600)
//DRLIPARM DD *
IMSVER=71 -- or IMSERVER=81
REPORTS=R2(FF)
IMSID=IMS1

//DRLSLOGP JOB (ACC000,001),'DRLSLOGP',
// NOTIFY=USERIDA,MSGCLASS=X,CLASS=A,REGION=0M
//DRLLOGP EXEC PGM=DRLSLOGP
//DD DISP=SHR,DSN=dr170.SDRLLOAD
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS1
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS2
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS3
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS4
//DD DISP=SHR,DSN=IMS SLODS/OLDS extract log from IMS5
//DD DUMMY -- or LRECL=32756 output composite data set
//DD DUMMY -- or LRECL=32756 output checkpoint data set
/*
//DRLOUT DD SYSOUT**,DCB=(RECFM=F,LRECL=80)
//DRLDUMP DD SYSOUT**,DCB=(RECFM=VB,LRECL=32756)
//DRLRPT2 DD SYSOUT**,DCB=(RECFM=FB,LRECL=600)
//DRLIPARM DD *
SQNLOGS=5
REPORTS=R2(FF)
IMSVER=71 -- or IMSERVER=81
IMSIDCHECK=CONTINUE
//

Figure 27. Sample jobs for running DRLSLOGP

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 (SLU P) 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.

Administering the IMS Shared Queue feature

- 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 z/OS 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 z/OS processing (COLLECT) and Tivoli Decision Support for z/OS 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.
Administering the IMS Shared Queue feature

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 z/OS is not installed, needs to be built and then redistributed. To build this Load Lib at the central location where Tivoli Decision Support for z/OS is installed, you can customize and use the sample job DRLJCSQL in the Tivoli Decision Support for z/OS CNTL library.

//DRLJIMSL JOB (ACCTE), 'IMS LIGHT LIB'

/******************************************************************************

*
* LICENSED MATERIALS - PROPERTY OF IBM
*
* 5695-101 (C) Copyright IBM CORPORATION 2003
* SEE COPYRIGHT INSTRUCTIONS
*
******************************************************************************

* NAME: DRLJCSQL
* STATUS: Tivoli Decision Support for z/OS 1.7.0
* FUNCTION:
* PREPARE LOAD LIBRARY TO RUN STANDALONE THE DRLSLOGP IMS LOG
* 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 TDS SMP INSTALLED LIBRARY.
* OUTLIB CONTAINS THE OUTPUT COPY LOAD LIBRARY
* - DECIDE WHETHER SMP DISTRIBUTION ON TARGET LIBRARIES ARE SUITABLE TO BE USED AS SOURCE AND UPDATE INLIB DSN ACCORDINGLY (SDRLLOAD OR ADRLLOAD)
* - DELETE THE SELECT STATEMENT ROW CORRESPONDING TO THE IMS RELEASE YOU ARE NOT INTERESTED IN, IF ANY:
* DRLZI71L,... FOR IMS V7R1M0
* DRLZI61L,... FOR IMS V8R1M0
* - DOUBLE CHECK IN THE JOB OUTPUT THAT ALL THE NEEDED LOAD MODULES HAVE BEEN CORRECTLY COPIED.

******************************************************************************

//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//INLIB DD DISP=(SHR,KEEP),UNIT=3390, VOL=SER=yyyyy, DSN=dr1170.SDRLLOAD
//OUTLIB DD DSN=dr1170.IMSLIGHT.LOAD,DISP=(NEW,CATLG), SPACE=(CYL,(1,1,5));UNIT=3390, VOL=SER=zzzzzz, DCL=(RECFM=U, BLKSIZE=32760,LRECL=0)
//SYSIN DD *
COPY I=INLIB,O=OUTLIB
SELECT MEMBER=(DRLSLOGP, DRL2LOGJ, DRLCSQJ, DRLCIO24, DRLPMSG)
SELECT MEMBER=(DRLS171L, DRLS1712, DRLS171C)
SELECT MEMBER=(DRLS181L, DRLS1812, DRLS181C)

******************************************************************************

Statement description

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

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 in 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 CSQLIGHT.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.

```//DRLILOGP JOB (ACCT£),'DRLSLOGP IMS' //****************************************************************************** //* LICENSED MATERIALS - PROPERTY OF IBM * //* 5695-101 (C) COPYRIGHT IBM CORPORATION 2004 * //* SEE COPYRIGHT INSTRUCTIONS. * //****************************************************************************** //* NAME: DRLILOGP * //* STATUS: Tivoli Decision Support for z/OS 1.7 * //* 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=drl170.CSQLIGHT.LOAD //STEPLIB DD DISP=SHR,DSN=dsn710.DSNLOAD //DRLLOG DD DISP=SHR,DSN=... INPUT IMS SLOG 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 SYOUT=**,DCB=(RECFM=F,LRECL=80) //DRLDUMP DD SYOUT=**,DCB=(RECFM=VB,LRECL=32756) //DRLIRPT2 DD DISP=SHR,DSN=... OUTPUT RECORDS FROM R2 REPORT/PROCEDURE //DRLIPARM DD * IMSID= -- IMS ID FOR THIS IMS SYSTEM IMSVER=71 -- 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 z/OS 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_V710_COLLECL
- COLLECT CSQ_V810_COLLECL
Administering the IMS Shared Queue feature
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 12 on page 138 explains the sections of the composite record:

<table>
<thead>
<tr>
<th>Short description</th>
<th>A descriptive section name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The actual name of the section.</td>
</tr>
<tr>
<td>Prefix</td>
<td>The one-character prefix for all data subsections and fields defined in the record section.</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>
</tr>
</tbody>
</table>
# IMS Shared Queue record definitions

## Table 12. Composite record sections

<table>
<thead>
<tr>
<th>Short description</th>
<th>Name</th>
<th>Prefix</th>
<th>Section contents and explanation</th>
</tr>
</thead>
</table>
| Origin Unit of Work         | OUOW  | OW     | It is the main key of transaction. It is allocated on arriving<br>
|                             |       |        | X'01' Message queue insert<br>
|                             |       |        | X'03' M.Q.I. destination an SMB<br>
|                             |       |        | X'03' Message queue enqueue<br>
|                             |       |        | X'5901' Aoi exit<br>
|                             |       |        | This is the anchor section for the PUOW sections<br>                                                                        |
| Process Unit of Work        | PUOW  | PW     | X'01'/X'03' Message queue insert<br>
|                             |       |        | X'35' Message queue enqueue<br>
|                             |       |        | X'31' Message queue GU<br>
|                             |       |        | X'36' Message queue dequeue<br>
|                             |       |        | X'37' Message commit/transfer<br>
|                             |       |        | X'38' Message failed<br>
|                             |       |        | X'5901' EMH input<br>
|                             |       |        | X'5903' EMH output<br>
|                             |       |        | X'5936' EMH dequeue<br>
|                             |       |        | X'5937' Fast Path commit<br>
|                             |       |        | X'5938' Fast Path failed<br>
|                             |       |        | All the PUOW sections relative to the same OUOW are chained together in one composite record.<br>
|                             |       |        | All the records carrying the same Process Unit of Work are chained together at the same PUOW section.<br>          |
| Program                     | PSB   | PB     | X'08' Program scheduled<br>
|                             |       |        | X'07' Program terminated<br>
|                             |       |        | These records are in sequence for the program scheduling that processed the input message.<br>
|                             |       |        | **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.<br>
|                             |       |        | Each PSB section has a link with a PUOW section in one compsite record.                                                     |
Chapter 12. IMS Shared Queue data tables and lookup tables

The Tivoli Decision Support for z/OS database is a collection of DB2 tables. Each table contains a fixed number of columns. The number of rows in each table varies with time, because of rows added by the collect function and because of database maintenance.

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

This 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 67. For information about the relationships between tables and between records and tables, see “IMS Shared Queue Collect feature data flow” on page 122.

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.
IMS Shared Queue data tables and lookup tables

Data tables

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

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>
<tr>
<td>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PERFORMANCE_GROUP</td>
<td>k</td>
<td>CHAR(1)                                                                 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>ORIGIN_IMS</td>
<td>k</td>
<td>CHAR(8)                                                                 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>k</td>
<td>CHAR(8)                                                                 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>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>ROUTING_CODE</td>
<td>k</td>
<td>CHAR(8)                                                                 Code used by EMH to enable transactions to be routed to programs within LBL. From X’590I’ record.</td>
</tr>
<tr>
<td>APPC_MODE_NAME</td>
<td></td>
<td>CHAR(8)                                                                 The destination Network ID for the APPC session.</td>
</tr>
<tr>
<td>APPC_NETID</td>
<td></td>
<td>CHAR(8)                                                                 The destination Network ID for the APPC session.</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>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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</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 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 ransactions 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 program-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 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>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, that 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:
- IMS_TRAN_QUEUE_Q: 10 days
- IMS_TRAN_QUEUE_D: 35 days

<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>Date when the activity occurred.</td>
</tr>
<tr>
<td>TIME</td>
<td>k 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>k 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>k 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>k 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>k CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</td>
</tr>
<tr>
<td>PERFORMANCE_GROUP</td>
<td>k CHAR(1)</td>
<td>The transaction type based on (DRLMINPUT+DRLMPROCE) 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>k 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_LGMSG_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_LGMSG_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_LGMSG_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_LGMSG_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_LGMSG_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_LGMSG_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 Shared Queue data tables and lookup tables**

**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 Name</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>k CHAR(8)</td>
<td>The name of the IMS transaction the user requested.</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>ORIGIN_IMS</td>
<td>k 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>k 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>k 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>
</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>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 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>
</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>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 TRAN_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.
IMS Shared Queue data tables and lookup tables

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>
</tbody>
</table>
| PERFORMANCE_GROUP    | CHAR(1)   | The transaction type based on (DRLINPUT+ DRLMPROCE) value ranges. It could be:
- \( F = \text{FAST} \) (0.0 - 0.7)
- \( G = \text{GOOD} \) (0.7 - 1.5)
- \( M = \text{MEDIUM} \) (1.5 - 3.0)
- \( L = \text{LOW} \) (3.0 - 10.0)
- \( S = \text{SLOW} \) (above 10)
| QUEUE_TYPE           | CHAR(12)  | Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED. |
| INPUT_MSG_SEC_AVG    | REAL      | Average time spent by an input message on that queue. Calculated as INPUT_MSG_SEC / INPUT_MSG. |
| MVS_SYSTEM_ID        | CHAR(4)   | MVS system ID. |
| OUTPUT_MSG_SEC_AVG   | REAL      | Average time spent by an output message on that queue. Calculated as OUTPUT_MSG_SEC / OUTPUT_MSG. |
| SYSPLEX_NAME         | CHAR(8)   | Sysplex name. |
| TRANS_RATE           | REAL      | Average transaction arrival rate over the interval. Calculated as TRANSACTIONS / 900 (for quarter-hourly) or TRANSACTIONS / 86400 (for daily). |

**Note:** In addition to the key and data columns described here, these views also contain all the data columns described in "IMS_TRAN_QUEUE_Q/ _D" on page 144.
IMS Shared Queue data tables and lookup tables

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>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:MM:SS. 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 DLRTOKEN.</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>k CHAR(8)</td>
<td>The name of the IMS transaction the user requested. From DLRTRNCD.</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 DLRNPSB.</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>
</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>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>APPLICATION_NAME</td>
<td>CHAR(18)</td>
<td>Application name. This is from the RESOURCE_TARGET_NM in IMS_AVAIL_RESOURCE 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 he count stored in the program termination record (record type X'07'). Calculated as Sum of DLRAPS.</td>
</tr>
<tr>
<td>DLI_CHKPT_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 he 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_ENQUEUES</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 DLREXCW. 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.</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 DLRGCM.</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_GCMD_CALLS</td>
<td>REAL</td>
<td>Number of DL/I GCMD 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 DLRRINIT.</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 DLRRINQV.</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>
</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>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>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 DLRPURGE.</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_ENQUEUE</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 DLRQCWAT.</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>DLIROLS_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 DLRSTSDQ.</td>
</tr>
<tr>
<td>DLI_TEST_ENQUEUE</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 DLRSTSNQ.</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 DLRSTSWT.</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_ENQUEUE</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>
</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>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>DLI_XRST_CALLS</td>
<td>REAL</td>
<td></td>
</tr>
<tr>
<td>DLIB_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>DLIB_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 DLRDLET.</td>
</tr>
<tr>
<td>DLIB_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>DLIB_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 DLRGHNP.</td>
</tr>
<tr>
<td>DLIB_GHU_CALLS</td>
<td>REAL</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'). Calculated as Sum of DLRGHU.</td>
</tr>
<tr>
<td>DLIB_GN_CALLS</td>
<td>REAL</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'). Calculated as Sum of DLRGN.</td>
</tr>
<tr>
<td>DLIB_GNP_CALLS</td>
<td>REAL</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'). Calculated as Sum of DLRGNP.</td>
</tr>
<tr>
<td>DLIB_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>DLIB_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>DLIB_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 DLRREPL.</td>
</tr>
<tr>
<td>DLIBCTL_DBIOS_SEC</td>
<td>REAL</td>
<td>The total elapsed time for DB 1/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>DLIBCTL_DBIOS</td>
<td>REAL</td>
<td>The total number of DB 1/Os for DBCTL, derived from the count stored in the program termination record (record type X'07'). Calculated as Sum of DLRIOCNT.</td>
</tr>
<tr>
<td>DLIBCTL_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>Column name</td>
<td>Data type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</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 DLRISMES.</td>
</tr>
<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: X'80' UOR END, X'10' CPI REGION, X'08' QUICK RESC, X'04' DBCTL THRD, X'02' BMP REGION, X'01' MPP REGION, X'22' JBP REGION, X'21' JMP REGION</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 Shared Queue data tables and lookup tables

**IMS_AVAILABILITY_T**

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

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_AVAIL_Resource lookup table.</td>
</tr>
<tr>
<td>RESOURCE_TYPE</td>
<td>k Char(8)</td>
<td>Resource Type. Possible values are: IMSSYS IMS system</td>
</tr>
<tr>
<td>INTERVAL_TYPE</td>
<td>k Char(3)</td>
<td>Interval type. Possible values are: ===, ====, =</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 Shared Queue data tables and lookup tables

**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:

<table>
<thead>
<tr>
<th>Table</th>
<th>Retention Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS_AVAILABILITY_D</td>
<td>45 days</td>
</tr>
<tr>
<td>IMS_AVAILABILITY_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>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>k Char(8)</td>
<td>Sysplex name.</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>k Char(4)</td>
<td>MVS system ID.</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>k Char(8)</td>
<td>The IMS subsystem ID.</td>
</tr>
<tr>
<td>RESOURCE_NAME</td>
<td>k Char(18)</td>
<td>Resource Name.</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>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**

This section describes the lookup table.

**IMS_AVAIL_RESOURCE**

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

<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>
<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>
<tr>
<td>RESOURCE_TYPE</td>
<td>Char(8)</td>
<td>Resource type: IMSSYS IMS system, IMSREG IMS region, IMSPGM IMS program, IMSTRAN IMS transaction. IMSPGM and IMSTRAN will be used for tracking application.</td>
</tr>
<tr>
<td>AVAIL_OBJ_PCT</td>
<td>Decimal(4,1)</td>
<td>Availability objective for the resource, in percentage.</td>
</tr>
<tr>
<td>RESOURCE_CHKPOINT</td>
<td>Integer</td>
<td>Resource checkpoint interval in seconds.</td>
</tr>
<tr>
<td>RESOURCE_TARGET_NM</td>
<td>Char(18)</td>
<td>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.</td>
</tr>
<tr>
<td>SCHEDULE_NAME</td>
<td>Char(8)</td>
<td>Schedule name to use for the resource. If nothing is specified, STANDARD is used as default.</td>
</tr>
</tbody>
</table>

**Example of Table Contents:**

<table>
<thead>
<tr>
<th>SYSPLEX_NAME</th>
<th>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>RESOURCE_SCHEDULE_NAME</th>
<th>AVAILABLE_OBJ</th>
<th>PCT</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>%</td>
<td>%</td>
<td>IMS71A</td>
<td>IMSSYS</td>
<td>---</td>
<td>3600</td>
<td>STANDARD</td>
<td>95.0</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>%</td>
<td>DSWM%</td>
<td>IMSREG</td>
<td>---</td>
<td>3600</td>
<td>STANDARD</td>
<td>95.0</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>%</td>
<td>PGM%</td>
<td>IMSPGM</td>
<td>APPL1</td>
<td>---</td>
<td>STANDARD</td>
<td>95.0</td>
<td></td>
</tr>
<tr>
<td>%</td>
<td>%</td>
<td>%</td>
<td>TRAN%</td>
<td>IMSTRAN</td>
<td>APPL1</td>
<td>---</td>
<td>STANDARD</td>
<td>95.0</td>
<td></td>
</tr>
</tbody>
</table>

160 Tivoli Decision Support for z/OS: IMS Performance Feature Guide and Reference
Mapping between Non-SQ and SQ DB2 Tables

In the Shared Queue Feature, some of the old DB2 tables have been removed and most of their fields have been distributed into new DB2 tables. For example, fields from IMS_TRANSACTION_H and IMS_USER_TRAN_H tables have been moved to the new DB2 table IMS_TRAN_H.

The DB2 tables that no longer exist are:
- IMS_APPLICATION_H
- IMS_APPLICATION_W
- IMS_SYSTEM_D
- IMS_SYSTEM_Q
- IMS_TRANSACTION_D
- IMS_TRANSACTION_H
- IMS_TRANSACTION_W
- IMS_USER_APPL_D
- IMS_USER_APPL_W
- IMS_USER_TRAN_D
- IMS_USER_TRAN_H
- IMS_USER_TRAN_W

The following sections describe how the DB2 tables and their fields have been re-organized. In the tables below, the first two columns list the old (non-SQ) fields and the corresponding R2 fields from which they are calculated; the remaining columns list the DB2 table they belong to, the new names of the fields, and the records from which they derive.

The fields that no longer exist are marked with (*). Most of these fields have been merged and re-organized using the TRANS_TYPE field. For example, the information that was contained in the old BMP_FP_ABORTS field is now in the new FP_ABORTS field when the first bit of TRANS_TYPE value is set to ‘B’.

**Table 13. IMS_TRANSACTION_H, D, W Field Remap**

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>DRL_DATE</td>
<td>IMS_TRAN_x</td>
<td>DATE</td>
<td>DRL_DATE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>TIME</td>
<td>DRL_TIME</td>
<td>IMS_TRAN_H</td>
<td>TIME</td>
<td>DRL_TIME from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>MVS_SYSTEM_ID</td>
<td>Runtime</td>
<td>IMS_TRAN_x</td>
<td>MVS_SYSTEM_ID</td>
<td>Runtime Parameter</td>
</tr>
<tr>
<td>IMS_SYSTEM_ID</td>
<td>DRL_IMSID</td>
<td>IMS_TRAN_x</td>
<td>ORIGIN_IMS</td>
<td>PROCESS_IMS</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>DRL_TRANT1</td>
<td>IMS_TRAN_x</td>
<td>TRANSACTION_NAME</td>
<td>DRL_TRANT1</td>
</tr>
<tr>
<td></td>
<td>DRL_SMBNM</td>
<td></td>
<td></td>
<td>DRL_TXNAM from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>TRANSACTION_NAME</td>
<td>DRL_TRANT1</td>
<td>IMS_PSB_ACCOUNT_X</td>
<td>TRANSACTION_NAME</td>
<td>DLRTRNCD from CSQ_Vxxx_ST07</td>
</tr>
</tbody>
</table>
### Table 13. IMS_TRANSACTION_H, D, W Field Remap (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM_NAME</td>
<td>DRL_TRANT1 DRL_PSNBNM</td>
<td>IMS_TRAN_x</td>
<td>PROGRAM_NAME</td>
<td>DRL_TRANT1 DRL_PGMNM from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PROGRAM_NAME</td>
<td>DRL_TRANT1 DRL_PSNBNM</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>PSB_NAME</td>
<td>DLRNPSB from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>PERIOD_NAME</td>
<td>DRL_DATE DRL_TIME (+MVS_SYSTEM_ID as parameters to the PERIOD FUNCTION)</td>
<td>IMS_TRAN_x</td>
<td>PERIOD_NAME</td>
<td>DRL_DATE DRL_TIME from CSQ_Vxxx_R2(+ SYSTEM_ID as parameters to the PERIOD FUNCTION)</td>
</tr>
<tr>
<td>TRANSACTION_CLASS</td>
<td>DRL_TRANT1 DRL_TXCLS</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_TYPE</td>
<td>DRL_PRTM DRL_IQTM</td>
<td>IMS_TRAN_x</td>
<td>PERFORMANCE_GROUP</td>
<td>DRLMINPUT DRLPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>IMS_APLID</td>
<td>Runtime Parameter</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>APPLICATION_NAME</td>
<td>RESOURCE_NAME field from IMS_AVAIL_RESOURCE lookup table</td>
</tr>
<tr>
<td>IMS_CTRL_REGION</td>
<td>Runtime Parameter</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>IMS_VERSION</td>
<td>Runtime Parameter</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TPI_COMMITS</td>
<td>DRLACOM</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TPI_CPU_SEC</td>
<td>DRLATIME</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TPI_DSPBCALLS</td>
<td>DRLADET</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TPI_ENDED</td>
<td>DRLAEND</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TPI_STARTED</td>
<td>DRLASTA</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_NETID</td>
<td>DRLANID</td>
<td>IMS_TRAN_x</td>
<td>APPC_NETID</td>
<td>DRLANID from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>APPC_MODE_NAME</td>
<td>DRLAMON</td>
<td>IMS_TRAN_x</td>
<td>APPC_MODE_NAME</td>
<td>DRLAMON from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>APPC_TIBQAB_T</td>
<td>DRLATQT</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>APPC_TIBQAB_A</td>
<td>DRLATQA</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>BMP_DS_LMSG_BYTES(*)</td>
<td>DRL_R2TYP DRL_LMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>BMP_DS_LMSG_ISRT(*)</td>
<td>DRL_R2TYP DRL_LMSGI</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_LGMSG_AVG</td>
<td>DRL_TRANT1 DRLMSGQL DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_DS_LMSG_BYTES(*)</td>
<td>DRL_R2TYP DRL_LMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>BMP_DS_SMSG_BYTES(*)</td>
<td>DRL_R2TYP DRL_SMSG</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>BMP_DS_SMSG_ISRT(*)</td>
<td>DRL_R2TYP DRL_SMSG</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_SHMSG_AVG</td>
<td>DRL_TRANT1 DRLMSGQS DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>Column name</td>
<td>IMS_Vxxx_R2 fields</td>
<td>New table</td>
<td>New column name</td>
<td>New CSQ record fields</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>------------------------------</td>
<td>----------------</td>
<td>-----------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>BMP_FF_ABORTS(*)</td>
<td>DRL_R2TYP DRL_FFABRT</td>
<td>IMS_TRAN_x</td>
<td>FF_ABORTS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_FF_COMMITS(*)</td>
<td>DRL_R2TYP DRL_FPCMTS</td>
<td>IMS_TRAN_x</td>
<td>FF_COMMITS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_FP_ABORTS(*)</td>
<td>DRL_R2TYP DRL_FFABRT</td>
<td>IMS_TRAN_x</td>
<td>FP_ABORTS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_FP_COMMITS(*)</td>
<td>DRL_R2TYP DRL_FPCMTS</td>
<td>IMS_TRAN_x</td>
<td>FP_COMMITS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_OUTPUT_MSGS(*)</td>
<td>DRL_R2TYP DRL_OUTMSG</td>
<td>IMS_TRAN_x</td>
<td>OUTPUT_CSQ OUTPUT_LOCAL</td>
<td>DRL_TXFLOW DRL_TXFLAG from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_PROCESS_SEC(*)</td>
<td>DRL_R2TYP DRL_PRTM</td>
<td>IMS_TRAN_x</td>
<td>PROCESS_SEC</td>
<td>DRL_TRAN1 DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>BMP_PROGRAMS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>It can be calculated using a count operator on IMS_PSB_ACCOUNT_x</td>
<td></td>
</tr>
<tr>
<td>DLIDB_CALLS</td>
<td>DRLCLCNT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_CALLS</td>
<td>SUM(DLRCMD) from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDBCTL_DBIOS</td>
<td>DRLIOCNT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDBCTL_DBIOS</td>
<td>DRLIOCNT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDBCTL_DBIOS_SEC</td>
<td>DRLTMEIO</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDBCTL_DBIOS_SEC</td>
<td>DRLTMEIO from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDBCTL_LOCK_SEC</td>
<td>DRLTMEPL</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDBCTL_LOCK_SEC</td>
<td>DRLTMEPL from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_DLET_CALLS</td>
<td>DRLDLET</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_DLET_CALLS</td>
<td>DLRDLET from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GHN_CALLS</td>
<td>DRLGHN</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GHN_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GHN_CALLS</td>
<td>DRLGHN</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GHN_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GHNP_CALLS</td>
<td>DRLGHN</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GHNP_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GHU_CALLS</td>
<td>DRLGHU</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GHU_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GN_CALLS</td>
<td>DRLGN</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GN_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GNP_CALLS</td>
<td>DRLGNP</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GNP_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_GU_CALLS</td>
<td>DRLGU1</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_GU_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_ISRT_CALLS</td>
<td>DRLISRT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_ISRT_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDB_REPL_CALLS</td>
<td>DRLREPL</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDB_REPL_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDC_GN_CALLS</td>
<td>DRLGNMES</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDC_GN_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDC_GU_CALLS</td>
<td>DRLGUMES</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDC_GU_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLIDC_ISRT_CALLS</td>
<td>DRLISMES</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDC_ISRT_CALLS</td>
<td>DLRGHN from CSQ_Vxxx_ST07</td>
</tr>
</tbody>
</table>
### Table 13. IMS_TRANSACTION_H, D, W Field Remap (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLIDC_PURGE_CALLS</td>
<td>DRLPUMES</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLIDC_PURGE_CALLS</td>
<td>DLRPUMES from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_EXCL_DEQUEUES</td>
<td>DRLEXCDQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_EXCL_DEQUEUES</td>
<td>DLREXCDQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_EXCL_ENQUEUES</td>
<td>DRLEXCNQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_EXCL_ENQUEUES</td>
<td>DLREXCNQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_EXCL_ENQWAIT</td>
<td>DRLEXCWT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_EXCL_ENQWAIT</td>
<td>DLREXCWT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_GCMD_CALLS</td>
<td>DRLGCMD</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_GCMD_CALLS</td>
<td>DLRGCMD from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_QCMD_DEQUEUES</td>
<td>DRLQCODQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_QCMD_DEQUEUES</td>
<td>DLRQCODQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_QCMD_ENQUEUES</td>
<td>DRLQCONQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_QCMD_ENQUEUES</td>
<td>DLRQCONQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_QCMD_ENQWAIT</td>
<td>DRLSUPWT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_QCMD_ENQWAIT</td>
<td>DLRSUPWT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLISQ6_ACCUM_SEC</td>
<td>DRL_SQ6TM</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLISQ6_ACCUM_SEC</td>
<td>DLRSQ6TM from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_TEST_DEQUEUES</td>
<td>DRLTSTDQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_TEST_DEQUEUES</td>
<td>DLRTSTDQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_TEST_ENQUEUES</td>
<td>DRLTSTNQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_TEST_ENQUEUES</td>
<td>DLRTSTNQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_TEST_ENQWAIT</td>
<td>DRLTSTWT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_TEST_ENQWAIT</td>
<td>DLRSUPWT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_UPDT_DEQUEUES</td>
<td>DRLSUPDQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_UPDT_DEQUEUES</td>
<td>DLRSUPDQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_UPDT_ENQUEUES</td>
<td>DRLSUPNQ</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_UPDT_ENQUEUES</td>
<td>DLRSUPNQ from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_UPDT_ENQWAIT</td>
<td>DRLSUPWT</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>DLI_UPDT_ENQWAIT</td>
<td>DLRSUPWT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>EMH_BUFFER_BYTES(*)</td>
<td>DRL_R2TYP1 DRL_EMHBY</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>EMH_DS_LMSG_BYTES(*)</td>
<td>DRL_R2TYP1 DRL_LMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>EMH_DS_LMSG_ISRT(*)</td>
<td>DRL_R2TYP1 DRL_SMSGI</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_LMSG_AVG</td>
<td>DRL_TRANT1 DRLMSQGL DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_DS_SMSG_BYTES(*)</td>
<td>DRL_R2TYP1 DRL_SMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>EMH_DS_SMSG_ISRT(*)</td>
<td>DRL_R2TYP1 DRL_SMSGI</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_LMSG_AVG</td>
<td>DRL_TRANT1 DRLMSQGIS DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_FF_ABORTS(*)</td>
<td>DRL_R2TYP1 DRL_FFABRT</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>FF_ABORTS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_FF_COMMITS(*)</td>
<td>DRL_R2TYP1 DRL_FFCOMTS</td>
<td>IMS_TRAN</td>
<td>FF_COMMITS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_FP_ABORTS(*)</td>
<td>DRL_R2TYP1 DRL_FFPABRT</td>
<td>IMS_TRAN</td>
<td>FP_ABORTS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
</tbody>
</table>
### Table 13. IMS_TRANSACTION_H, D, W Field Remap (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMH_FP_COMMITS(*)</td>
<td>DRL_R2TYP1 DRI_FFPCMTS</td>
<td>IMS_TRAN_x</td>
<td>FP_COMMITS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_INPUT_SEC(*)</td>
<td>DRL_R2TYP1 DRI_IQTM</td>
<td>IMS_TRAN_x</td>
<td>INPUT_SEC(*)</td>
<td>DRL_TRANT1 DRLMINPUT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_MSGQ_OUTPUTS (*)</td>
<td>DRL_R2TYP1 DRI_OUTMSG</td>
<td>IMS_TRAN_x</td>
<td>OUTPUT_CSQ</td>
<td>DRL_TXFLOW DRL_TXFLAG from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_NETWORK_SEC (*)</td>
<td>DRL_R2TYP1 DRI_NETTM</td>
<td>IMS_TRAN_x</td>
<td>NETWORK_SEC</td>
<td>DRL_TRANT1 DRLMNETW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_OUTPUT_SEC(*)</td>
<td>DRL_R2TYP1 DRI_OQTM</td>
<td>IMS_TRAN_x</td>
<td>OUTPUT_SEC</td>
<td>DRL_TRANT1 DRLMOUTPUT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_PROCESS_SEC(*)</td>
<td>DRL_R2TYP1 DRI_PRTM</td>
<td>IMS_TRAN_x</td>
<td>PROCESS_SEC</td>
<td>DRL_TRANT1 DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_RESPONSE_SEC (*)</td>
<td>DRL_R2TYP1 DRI_TRNTM DRI_NETTM</td>
<td>IMS_TRAN_x</td>
<td>RESPONSE_SEC</td>
<td>DRL_TRANT1 DRLMRESP DRLMTRANS DRLMPROCE DRLMOUTPUT DRLMNETW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_1(*)</td>
<td>DRL_R2TYP1 DRI_TRNTM</td>
<td>IMS_TRAN_x</td>
<td>TRAN_CNTR_1</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_2(*)</td>
<td>DRL_R2TYP1 DRI_TRNTM</td>
<td>IMS_TRAN_x</td>
<td>TRAN_CNTR_2</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_3(*)</td>
<td>DRL_R2TYP1 DRI_TRNTM</td>
<td>IMS_TRAN_x</td>
<td>TRAN_CNTR_3</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRAN_CNTR_4(*)</td>
<td>DRL_R2TYP1</td>
<td>IMS_TRAN_x</td>
<td>TRAN_CNTR_4</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRANSACTIONS (*)</td>
<td>DRL_R2TYP1</td>
<td>IMS_TRAN_x</td>
<td>TRANSACTIONS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>EMH_TRANSIT_SEC(*)</td>
<td>DRL_R2TYP1 DRI_TRNTM</td>
<td>IMS_TRAN_x</td>
<td>TRANSIT_SEC</td>
<td>DRL_TRANT1 DRLMTRANS DRLMINPUT DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_CI_HNH_CONT</td>
<td>DRL_R2TYP1 DRLFHCCCT DRL_TRANT</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
</tbody>
</table>
### Table 13. IMS_TRANSACTION_H, D, W Field Remap (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP_CI_NHNH_CONT</td>
<td>DRL_R2TYP1 DRLFUOWC DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_CI_NHNH_CONT</td>
<td>DRL_TXFLOW DRLFUOWC from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_COMBINATIONS</td>
<td>DRL_R2TYP1 DRLFCOMB DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_COMBINATIONS</td>
<td>DRL_TXFLOW DRLFCOMB from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_DEDB_BFR_Waits</td>
<td>DRL_R2TYP1 DRLFBFWT DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_DEDB_BFR_Waits</td>
<td>DRL_TXFLOW DRLFBFWT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_DEDB_CALLS</td>
<td>DRL_R2TYP1 DRLFDECL DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_DEDB_CALLS</td>
<td>DRL_TXFLOW DRLFDECL from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_DEDB_HSSP_CALLS</td>
<td>DRL_R2TYP1 DRLFHCAL DRL_TRANT</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>FP_DEDB_HSSP_PUTS</td>
<td>DRL_R2TYP1 DRLFHPI DRL_TRANT</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>FP_DEDB_NH_PUTGETS</td>
<td>DRL_R2TYP1 DRLFDERD DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_DEDB_NH_PUTGETS</td>
<td>DRL_TXFLOW DRLFDERD from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_LOGGED_CI</td>
<td>DRL_R2TYP1 DRLFLGCI DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_LOGGED_CI</td>
<td>DRL_TXFLOW DRLFLGCI from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_MSDB_CALLS</td>
<td>DRL_R2TYP1 DRLFMSCL DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_MSDB_CALLS</td>
<td>DRL_TXFLOW DRLFMSCL from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_OVERFLOW_BFR</td>
<td>DRL_R2TYP1 DRLFOVFN DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>FP_OVERFLOW_BFR</td>
<td>DRL_TXFLOW DRLFOVFN from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>FP_UOW_HNH_CONT</td>
<td>DRL_R2TYP1 DRLFHUCT DRL_TRANT</td>
<td>IMS_TRAN_x</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>FP_UOW_NHNH_CONT</td>
<td>DRL_R2TYP1 DRLFNUCT DRL_TRANT</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_CONV_STARTED</td>
<td>DRL_R2TYP1 DRL_TRANT3</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_CONV.TRAN</td>
<td>DRL_R2TYP1 DRL_TRANT3</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_DS_LMSG_BYTES</td>
<td>DRL_R2TYP1 DRL_LMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_DS_LMSG_ISRRT (*)</td>
<td>DRL_R2TYP1 DRL_LMSGB</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_LMSG_AVG INP_MSGQ_LMSG_MAX INP_MSGQ_LMSG_MIN OUT_MSGQ_LMSG_AVG OUT_MSGQ_LMSG_MIN</td>
<td>DRL_TRANT1 DRLIMSQL DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>MSGQ_DS_SMSG_BYTES</td>
<td>DRL_R2TYP1 DRL_SMSGB</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_DS_SMSG_ISRRT (*)</td>
<td>DRL_R2TYP1 DRL_SMSGI</td>
<td>IMS_TRAN_QUEUE_x</td>
<td>INP_MSGQ_SMSG_AVG INP_MSGQ_SMSG_MAX INP_MSGQ_SMSG_MIN OUT_MSGQ_SMSG_AVG OUT_MSGQ_SMSG_MAX OUT_MSGQ_SMSG_MIN</td>
<td>DRL_TRANT1 DRLOMSQL DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
</tbody>
</table>

**Note:**
- (*) Fields are not available for the specified transaction types.
<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGQ_DS_SPA_ISRT</td>
<td>DRL_R2TYP</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_FF_ABORTS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>FF_ABORTS</td>
<td>DRL_TXFLOW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_FF_COMMITS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>FF_COMMITS</td>
<td>DRL_TXFLOW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_FP_ABORTS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>FP_ABORTS</td>
<td>DRL_TXFLOW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_FP_COMMITS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>FP_COMMITS</td>
<td>DRL_TXFLOW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_INPUT_MSGS(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>INPUT_CSQ INPUT_LOCAL</td>
<td>DRL_TXFLOW DRL_TXFLAG from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_INPUT_SEC(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>INPUT_SEC</td>
<td>DRL.TRANT1 DRLMINPUT from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_MSG_SWITCHES</td>
<td>DRL_R2TYP</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>MSGQ_NETWORK_SEC (*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>NETWORK_SEC</td>
<td>DRL.TRANT1 DRLNETW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_OUTPUT_MSGS (*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>OUTPUT_CSQ OUTPUT_LOCAL</td>
<td>DRL_TXFLOW DRL_TXFLAG from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_OUTPUT_SEC(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>OUTPUT_SEC</td>
<td>DRL.TRANT1 DRLMOUTPUT from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_PROCESS_SEC (*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>PROCESS_SEC</td>
<td>DRL.TRANT1 DRLMPROC from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_RESPONSES(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>RESPONSES</td>
<td>DRL_TXFLOW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_RESPONSE_SEC(*)</td>
<td>DRL_R2TYP</td>
<td>IMS.TRAN_x</td>
<td>RESPONSE_SEC</td>
<td>DRL_TRANT1 DRLMRESP DRLMTRANS DRLMPROC DRLMOUTPUT DRLMNETW from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_1 (*)</td>
<td>DRL_R2TYP</td>
<td>IMS.SYSTEM.TRAN_x</td>
<td>TRAN_CNTR_1</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROC from CSQ.Vxxx.R2</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_2 (*)</td>
<td>DRL_R2TYP</td>
<td>IMS.SYSTEM.TRAN_x</td>
<td>TRAN_CNTR_2</td>
<td>DRL_TXFLOW DRL_TRANT1 DRLMINPUT DRLMPROC from CSQ.Vxxx.R2</td>
</tr>
</tbody>
</table>
### IMS Shared Queue data tables and lookup tables

#### Table 13. IMS_TRANSACTION_H, D, W Field Remap (continued)

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSGQ_TRAN_CNTR_3 (*)</td>
<td>DRL_R2TYP, DRL_TRNTM</td>
<td>IMS_SYSTEM_TRAN_x</td>
<td>TRAN_CNTR_3</td>
<td>DRL_TXFLOW, DRL_TRANT1, DRLMINPUT, DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>MSGQ_TRAN_CNTR_4 (*)</td>
<td>DRL_R2TYP, DRL_TRNTM</td>
<td>IMS_SYSTEM_TRAN_x</td>
<td>TRAN_CNTR_4</td>
<td>DRL_TXFLOW, DRL_TRANT1, DRLMINPUT, DRLMPROCE from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>MSGQ_TRANSACTIONS (*)</td>
<td>DRL_R2TYP</td>
<td>IMS_TRAN_x</td>
<td>TRANSACTIONS</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>MSGQ_TRANSIT_SEC(*)</td>
<td>DRL_R2TYP, DRL_TRNTM</td>
<td>IMS_TRAN_x</td>
<td>TRANSIT_SEC</td>
<td>DRL_TRANT1, DRLMTTRANS, DRLMINPUT, DRLMPROCE, DRLMOUTPUT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PGM_CPU_SEC</td>
<td>DRLTIME</td>
<td>IMS_TRAN_x</td>
<td>PGM_CPU_APPROX</td>
<td>DRL_TXFLOW, DRLCPUTOT, DRLTRATOT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>PGM_CPU_SEC</td>
<td>DLRTIME</td>
</tr>
<tr>
<td>PGM_FAILURES</td>
<td>DRL_ABEND</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>PGM_FAILURES</td>
<td>DLRCOMP from CSQ_Vxxx_ST107 and L0ACOMP from CSQ_Vxxx_ST10A07</td>
</tr>
<tr>
<td>PGM_TIMER_UNITS</td>
<td>DRLTIME</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_BNDY1_SEC</td>
<td>Fixed value (1.0)</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_BNDY2_SEC</td>
<td>Fixed value (2.0)</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_BNDY3_SEC</td>
<td>Fixed value (5.0)</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
<tr>
<td>RESPONSE_BNDY4_SEC</td>
<td>Fixed value (10.0)</td>
<td>Not Available</td>
<td>Not Available</td>
<td></td>
</tr>
</tbody>
</table>

#### IMS_USER_TRAN_x (H, D, W)

#### Table 14. IMS_USER_TRAN_x (H, D, W) Field Remap

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER_ID</td>
<td>DRL_OlTERM1, DRL_TRANT1, DRL_OUSER1, DRL_OUSER, DRL_OlTERM</td>
<td>IMS_TRAN_x</td>
<td>USER_ID</td>
<td>DRL_OlTERM1, DRL_TRANT1, DRL_OUSER1, DRL_OUSER, DRL_OlTERM from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>LOGICAL_TERMINAL</td>
<td>DRL_OlTERM</td>
<td>Not Available</td>
<td>ORIGIN_LTERM, DESTINATION_LTERM</td>
<td>DRL_TRANT1, DRL_OlTERM, DRL_DLTERM from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PHYSICAL_TERMINAL</td>
<td>DRL_TRANT1, DRL_OlTERM1, DRL_OLINE, DRL_OlTERM, DRL_OlTERM</td>
<td>IMS_TRAN_x</td>
<td>Not Available</td>
<td>Runtime Parameter</td>
</tr>
</tbody>
</table>
IMS System (Q, D)

Table 15. IMS System (Q, D) Field Remap

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION_JOB_NAME</td>
<td>DRL_TRANT1</td>
<td>IMS_SYSTEM_TRAN_x</td>
<td>REGION_JOB_NAME</td>
<td>DRL_TRANT1, DRL_REGION1, DRL_REGION,</td>
</tr>
<tr>
<td></td>
<td>DRL_REGION1</td>
<td></td>
<td></td>
<td>from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td></td>
<td>DRL_REGION</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| REGION_PST_ID  | DRL_TRANT1         | IMS_SYSTEM_TRAN_x  | PST_ID          | DRL_PSTID from CSQ_Vxxx_R2                |
|                | DRL_DRID           |                    |                 |                                           |

IMS Application (H, W)

Table 16.

<table>
<thead>
<tr>
<th>Column name</th>
<th>IMS_Vxxx_R2 fields</th>
<th>New table</th>
<th>New column name</th>
<th>New CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION_NAME</td>
<td>It is derived from the IMS_APPLICATION lookup table.</td>
<td>IMS_PSB_ACCOUNT_x</td>
<td>APPLICATION_NAME</td>
<td>It is derived from the IMS_AVAIL_RESOURCE lookup table.</td>
</tr>
</tbody>
</table>

IMS User Appl (D, W)

All the columns have already been described in the previous tables.

Mapping between New DB2 Table Fields and CSQ Records

The following sections map some of the columns of the new tables with the CSQ records from which they derive.

IMS Transaction (H, D, W)

Table 17. IMS Transaction (H, D, W) Description

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESTINATION_LTERM</td>
<td>The IMS-defined logical name for the terminal used to receive the transaction output. Missing if APFC OTMA.</td>
<td>It is derived from the IMS_AVAILRESOURCE lookup table.</td>
</tr>
<tr>
<td>ROUTING_CODE</td>
<td>Code used by EMH to enable transactions to be routed to programs within LBL. From X5901 record.</td>
<td>DRL_TRANT1, DRL_ROUTCD from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PGM_SWITCHES</td>
<td>Number of program-to-program switches.</td>
<td>DRLPGMWSW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PGM_SWITCHES_SEC</td>
<td>It is the time from the x07 of the root/children transaction and the x08 of the children/root transaction.</td>
<td>DRL_TRANT1, DRLPGMWSW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>SQ6_TIME</td>
<td>The total transaction time for subqueue 6, in seconds, as stored in the DL/I GU (record type X31) and program termination(record type X07) records. This represents the total time spent waiting in a wait-for-input or pseudo wait-for-input region with no work to do.</td>
<td>DRLSQ6TIME from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>TRANS_PRIOR</td>
<td>Message priority.</td>
<td>DRL_TXPRI</td>
</tr>
</tbody>
</table>
### IMS_SHARED_QUEUE_x (Q, D)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUEUE_TYPE</td>
<td>Queue Type. It can be: MSGQ LOCAL, EMHQ LOCAL, MSGQ SHARED.</td>
<td>DRL_TRANT1, DRL_TRANT2 from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>INPUT_MSG</td>
<td>The total number of input processed.</td>
<td>DRL_TXFLOW from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>INPUT_MSG_SEC</td>
<td>The total time, in seconds, the input messages spent on that queue</td>
<td>DRL_TRANT1, DRL_MIN_INPUT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>INP_EMHQ_MAX</td>
<td>Maximum number of the messages on the input queue for fast path transactions present before processing.</td>
<td>DRL_TRANT1, DRL_EMHQ from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>INP_EMHQ_MIN</td>
<td>Minimum number of the messages on the input queue for fast path transactions present before processing.</td>
<td>DRL_TRANT1, DRL_EMHQ from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>OUTPUT_MSG</td>
<td>The total number of output processed.</td>
<td>DRL_TXFLAG from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>OUTPUT_MSG_SEC</td>
<td>The total time, in seconds, the output messages spent on that queue</td>
<td>DRL_TRANT1, DRL_MOUTUT from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>OUT_EMHQ_AVG</td>
<td>Average number of the messages on the output queue for fast path transactions present before processing.</td>
<td>DRL_TRANT1, DRL_EMHQ from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>OUT_EMHQ_MAX</td>
<td>Maximum number of the messages on the output queue for fast path transactions present before processing.</td>
<td>DRL_TRANT1, DRL_EMHQ from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>OUT_EMHQ_MIN</td>
<td>Minimum number of the messages on the output queue for fast path transactions present before processing.</td>
<td>DRL_TRANT1, DRL_EMHQ from CSQ_Vxxx_R2</td>
</tr>
</tbody>
</table>

### IMS_SYSTEM_TRAN_x (H, D)

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGM_SWITCHES</td>
<td>Number of program-to-program switches.</td>
<td>DRL_PGMSWN from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>PGM_SWITCHES_SEC</td>
<td>It is the time from the x07 of the root/children transaction and the x08 of the children/root transaction.</td>
<td>DRL_PGMS from CSQ_Vxxx_R2</td>
</tr>
<tr>
<td>SQ6_TIME</td>
<td>The total transaction time for subqueue 6, in seconds, as stored in the DL/I GU (record type X31) and program termination (record type X07) records. This represents the total time spent waiting in a wait-for-input or pseudo wait-for-input region with no work to do.</td>
<td>DRLSQ6TIME from CSQ_Vxxx_R2</td>
</tr>
</tbody>
</table>
IMS Shared Queue data tables and lookup tables

**IMS_PSB_ACCOUNT_x (H, D, W)**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column description</th>
<th>CSQ record fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLI_APSB_CALLS</td>
<td>Number of DL/I APSB calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRAPSB.</td>
<td>DLRAPSB from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_CHKP_CALLS</td>
<td>Number of DL/I CHKP calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRCHKP.</td>
<td>DLRCHECKP from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_DPSB_CALLS</td>
<td>Number of DL/I DPSB calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRDPSB.</td>
<td>DLRDPSB from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_GMSG_CALLS</td>
<td>Number of DL/I GMSG calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRGMSG.</td>
<td>DLRGMSG from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_ICMD_CALLS</td>
<td>Number of DL/I ICMD calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRICMD.</td>
<td>DLRICMD from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_RCMD_CALLS</td>
<td>Number of DL/I RCMD calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRRCMD.</td>
<td>DLRRCMD from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_xRST_CALLS</td>
<td>Number of DL/I XRST calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRXRST.</td>
<td>DLRXRST from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_ROLB_CALLS</td>
<td>Number of DL/I ROLB calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLROLB.</td>
<td>DLROLB from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_ROLS_CALLS</td>
<td>Number of DL/I ROLS calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLROLS.</td>
<td>DLROLS from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_SETS_CALLS</td>
<td>Number of DL/I SETS calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSETS.</td>
<td>DLRSETS from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_SETU_CALLS</td>
<td>Number of DL/I SETU calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSETU.</td>
<td>DLRSETU from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_INIT_CALLS</td>
<td>Number of DL/I INIT calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRINIT.</td>
<td>DLRINIT from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_INQY_CALLS</td>
<td>Number of DL/I INQY calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRINQY.</td>
<td>DLRINQY from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>DLI_SLOG_CALLS</td>
<td>Number of DL/I SLOG calls, derived from the count stored in the program termination record (record type X07). Calculated as Sum of DLRSLOG.</td>
<td>DLRSLOG from CSQ_Vxxx_ST07</td>
</tr>
<tr>
<td>TRANS_PRIOR</td>
<td>TRANS_PRIOR CHAR(2) priority.</td>
<td>1.0APRTY from CSQ_Vxxx_ST07</td>
</tr>
</tbody>
</table>

**New IMS CSQ Composite Record Header Layout (DRLCSQCR)**

This record built by IMS CSQ Logproc (DRLSIxxL) is the input for IMS CSQ record procedure (DRLSIxx2). It can be saved in the DRLICOMP data set by the DRLSLOGP utility.

Declare

```
1 comp
2 fill_length
2 zz
```

Based(p_result) Boundary(Word),

Fixed Binary(15) Unsigned,

Fixed Binary(15) Unsigned,
IMS Shared Queue data tables and lookup tables

2 prefix Boundary(Word),
3 log_type Bit(08),
3 sub_type Bit(08),
3 tran_type Char(08) Boundary(Hword),
4 region_type Bit(08),
4 queue_type Bit(08),
4 data_comm Bit(08),
4 thread_mgm Bit(08),
4 pgm_switch Bit(08),
4 mixed_mode Bit(08),
4 envir_type Bit(08),
4 comp_status Bit(08),
3 * Char(02),
3 ims_id Char(08),
3 last_found_time Char(08),
3 pseudo_deq_time Char(08),
3 pseudo_deq_sq6_time Char(04),
3 smb_data Char(12) Boundary(Word),
4 smbname Char(08),
4 txnclass Char(01),
4 txnpriority Char(01),
4 proclim_cnt Fixed Binary(15) Unsigned,
3 ouow_section_offset Fixed Binary(31) Unsigned,
3 ouow_section_ll Fixed Binary(15) Unsigned,
3 no_of_ouow_section Fixed Binary(15) Unsigned,
3 puow_section_offset Fixed Binary(31) Unsigned,
3 puow_section_ll Fixed Binary(15) Unsigned,
3 no_of_puow_section Fixed Binary(15) Unsigned,
3 psb_section_offset Fixed Binary(31) Unsigned,
3 psb_section_ll Fixed Binary(15) Unsigned,
3 no_of_psb_section Fixed Binary(15) Unsigned;

Declare
comp_rec_data Char(32760) Based;

Mapping between SQ and non-SQ IMS R2 Fields

Table 21. SQ and non-SQ IMS R2 Field Remap

<table>
<thead>
<tr>
<th>CSQ_Vxxx_R2</th>
<th>CSQ_Vxxx_R2 field description</th>
<th>IMS_Vxxx_R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL_TRANT</td>
<td>The type of IMS transaction event</td>
<td>DRL_TRANT (different meaning)</td>
</tr>
<tr>
<td>DRL_DATE</td>
<td>First date found for the transaction</td>
<td>DRL_DATE</td>
</tr>
<tr>
<td>DRL_TIME</td>
<td>First time found for the transaction</td>
<td>DRL_TIME</td>
</tr>
<tr>
<td>DRL_OLETERM</td>
<td>The origin LTERM</td>
<td>DRL_OLETERM</td>
</tr>
<tr>
<td>DRL_OIMS</td>
<td>The origin IMS</td>
<td>DRL_OIMS</td>
</tr>
<tr>
<td>DRL_PIMS</td>
<td>The processing IMS</td>
<td>DRL_PIMS</td>
</tr>
<tr>
<td>DRL_OUSER</td>
<td>The origin USERID</td>
<td>DRL_OUSER</td>
</tr>
<tr>
<td>DRL_DLTERM</td>
<td>The destination LTERM</td>
<td>DRL_DLTERM</td>
</tr>
<tr>
<td>DRL_TXNAM</td>
<td>The transaction name</td>
<td>DRL_TXNAM</td>
</tr>
<tr>
<td>DRL_PGMNM</td>
<td>The program name</td>
<td>DRL_PGMNM</td>
</tr>
<tr>
<td>DRL_REGION</td>
<td>The dependent region job name</td>
<td>DRL_REGION</td>
</tr>
<tr>
<td>DRL_ROUTCD</td>
<td>The routing code</td>
<td>N/A</td>
</tr>
<tr>
<td>DRL_PSTID</td>
<td>Transaction Pst Id</td>
<td>DRL_PSTID</td>
</tr>
<tr>
<td>DRL_TXPRI</td>
<td>Transaction priority</td>
<td>DRL_TXPRI</td>
</tr>
<tr>
<td>DRL_TXFLOW</td>
<td>Internal</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Table 21. SQ and non-SQ R2 Field Remap (continued)

<table>
<thead>
<tr>
<th>CSQ_Vxxx_R2</th>
<th>CSQ_Vxxx_R2 field description</th>
<th>IMS_Vxxx_R2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRL_TXF.flag</td>
<td>Internal</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLFUUW.C</td>
<td>Number of CI contentsions between NH/NH EPCBs</td>
<td>DRLFUUW.C</td>
</tr>
<tr>
<td>DRLFCOMB</td>
<td>Count of combinations during logging 5950's</td>
<td>DRLFCOMB</td>
</tr>
<tr>
<td>DRLFDECL</td>
<td>Number of DEDB calls</td>
<td>DRLFDECL</td>
</tr>
<tr>
<td>DRLFDERD</td>
<td>Number of DEDB puts/gets</td>
<td>DRLFDERD</td>
</tr>
<tr>
<td>DRLFLGCI</td>
<td>Count of whole CIs logged</td>
<td>DRLFLGCI</td>
</tr>
<tr>
<td>DRLFQVFN</td>
<td>Number of overflow buffers used</td>
<td>DRLFQVFN</td>
</tr>
<tr>
<td>DRLFBFWT</td>
<td>Number of waits for DEDB buffers</td>
<td>DRLFBFWT</td>
</tr>
<tr>
<td>DRLFMSCL</td>
<td>Number of MSDB calls</td>
<td>DRLFMSCL</td>
</tr>
<tr>
<td>DRLMINUT</td>
<td>Input sec</td>
<td>DRL_QUERY</td>
</tr>
<tr>
<td>DRLMICSQ</td>
<td>Input csq sec</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLMPROCE</td>
<td>Process time</td>
<td>DRL_PROCESS</td>
</tr>
<tr>
<td>DRLPGMSWS</td>
<td>PGM Switch sec</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLPGMSWN</td>
<td>PGM SWITCH num</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLMOUTPUT</td>
<td>Output sec</td>
<td>DRL_OUTPUT</td>
</tr>
<tr>
<td>DRLMOCSQ</td>
<td>Output csq time</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLSQ6TME</td>
<td>Subqueue 6 time</td>
<td>DRL_SUBQ</td>
</tr>
<tr>
<td>DRLMNETW</td>
<td>Network time</td>
<td>DRL_NET</td>
</tr>
<tr>
<td>DRLMTRANS</td>
<td>Transit time</td>
<td>DRL_TRNS</td>
</tr>
<tr>
<td>DRLMRESP</td>
<td>Response sec</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLCPU.TOT</td>
<td>CPU TOTAL PSB</td>
<td>DRL_CPU</td>
</tr>
<tr>
<td>DRLTRATOT</td>
<td>NTRAN TOTAL PSB</td>
<td>DRL_TRAN</td>
</tr>
<tr>
<td>DRLANID</td>
<td>APPC Network ID</td>
<td>DRLANID</td>
</tr>
<tr>
<td>DRLAMON</td>
<td>APPC Mode Name</td>
<td>DRLAMON</td>
</tr>
<tr>
<td>DRLIEMHQ.M</td>
<td>Input EMH Queue msg cnt</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLIMSQQ5</td>
<td>Input MSGQ Queue short msg cnt</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLIMSQL</td>
<td>Input MSGQ Queue long msg cnt</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLOEMHQ.M</td>
<td>Output EMH Queue msg cnt</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLOMSQQ5</td>
<td>Output MSGQ Queue short msg cnt</td>
<td>N/A</td>
</tr>
<tr>
<td>DRLOMSQQL</td>
<td>Output MSGQ Queue long msg cnt</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Note:** The old IMS_Vxxx_R2 fields from the IMS x’07’ record (DLI section) are no longer in the CSQ_Vxxx_R2 record. They are available in the new CSQ_V710_ST07 record (used to populate the IMS_PSB_ACCOUNT table).
Part 3. Appendixes
Appendix A. Reports

Tivoli Decision Support for z/OS produces reports based on the data in the Tivoli Decision Support for z/OS database. Reports can show data from tables or from views. You can request reports using the Tivoli Decision Support for z/OS 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 z/OS feature reports:
- Report title
- Report ID
- Report group
- Source
- Attributes
- Variables
- Report types
- Standard report formats

Report title

Each report has a title. Each report title begins with an abbreviation that identifies the component. IMS 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 z/OS 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: Performance control task
  - Service: Service level planning task
  - Capacity: Capacity planning task
  - Security: Security control task
  - Configuration: Configuration management discipline
  - Operation: Operations management discipline
  - Change: Change management discipline
  - Problem: Problem management discipline

These are also specified where appropriate:

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

Variables

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

Report types

The IMS 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.

**Graphic reports**

In some cases, the meaning of data is best presented in graphic form. Graphic reports in Tivoli Decision Support for z/OS have both a QMF format and a Graphical Data Display Manager (GDDM®) format. Figure 29 on page 180 illustrates a graphic report.
Samples of reports across non-SQ and SQ tables

If you decide to use:

- Tivoli Decision Support for z/OS shared queue engine for IMS 7.1 not running in shared queue
- Tivoli Decision Support for z/OS shared queue engine for IMS 7.1 running in shared queue
- Tivoli Decision Support for z/OS 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 181.
This query uses the following fields from the IMS_SYSTEM_D pre-SQ table:

- **MVS_SYSTEM_ID**
- **IMS_SYSTEM_ID**
- **EMH_TRAN_CNTR_x** \(1 \leq x \leq 4\)
- **EMH_TRANSACTIONS**
- **MSGQ_TRAN_CNTR_x** \(1 \leq x \leq 4\)
- **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 182**.
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

```
SELECT
  MVS_SYSTEM_ID,
  IMS_SYSTEM_ID,
  DATE,
  VALUE((SUM(EMH_TRAN_CNTR_1) + SUM(MSGQ_TRAN_CNTR_1)) / (SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100,
  VALUE((SUM(EMH_TRAN_CNTR_2) + SUM(MSGQ_TRAN_CNTR_2)) / (SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100,
  VALUE((SUM(EMH_TRAN_CNTR_3) + SUM(MSGQ_TRAN_CNTR_3)) / (SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100,
  VALUE((SUM(EMH_TRAN_CNTR_4) + SUM(MSGQ_TRAN_CNTR_4)) / (SUM(EMH_TRANSACTIONS) + SUM(MSGQ_TRANSACTIONS))), 0)*100
FROM &PREFIX.IMS_SYSTEM_D - - PRE SQ TABLE
WHERE
  MVS_SYSTEM_ID = &MVS_SYSTEM_ID
  AND IMS_SYSTEM_ID = &IMS_SYSTEM_ID
  AND DATE >= &PRESQ_FROM_DATE
  AND DATE <= &PRESQ_TO_DATE
GROUP BY
  MVS_SYSTEM_ID,
  IMS_SYSTEM_ID,
  DATE
UNION
SELECT
  MVS_SYSTEM_ID,
  ORIGIN_IMS,
  DATE,
  VALUE((SUM(TRAN_CNTR_1)) / (SUM(TRANSACTIONS))), 0)*100,
  VALUE((SUM(TRAN_CNTR_2)) / (SUM(TRANSACTIONS))), 0)*100,
  VALUE((SUM(TRAN_CNTR_3)) / (SUM(TRANSACTIONS))), 0)*100,
  VALUE((SUM(TRAN_CNTR_4)) / (SUM(TRANSACTIONS))), 0)*100
FROM &PREFIX.IMS_SYSTEM_TRAN_D
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
z/OS engine, and from January, 15th you started using the new Tivoli Decision Support for z/OS SQ engine, when you are prompted for the input variables in the data selection panel (DRLDRSEL), 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 z/OS 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
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 z/OS database 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

![IMS Application Transaction Overview graphic report](image)

*Figure 35. Example of IMS Application Transaction Overview graphic report*

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).
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
Reports

**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:

- **Report ID**: IMSA03
- **Report group**: IMS reports
- **Source**: IMS_APPLICATION_W
- **Attributes**: IMS, performance, utilization, application, trend
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, APPLICATION_NAME, FROM_DATE, TO_DATE

![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 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).
Reports

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

![Image of IMS System Transaction Volumes Trend 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](image)

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 DLI 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

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 DLI database calls, derived from the count stored in the program termination record (record type X'07'). This represents the approximate total number of DLI 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

![Report Example](image-url)  

**Figure 43. Example of IMS User ID Response Time and CPU Detail by Date tabular report**

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.
Reports

**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:

- **Report ID**: IMST01
- **Report group**: IMS reports
- **Source**: IMS_TRANSACTION_D
- **Attributes**: IMS, transaction, utilization, performance, daily
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, IMS_TRANSACTION_NAME, DATE

## IMS Transaction Utilization Detail by Date report

**Date**: 2000-02-26  
**System**: 'MVS2'  
**IMS System**: 'IMS2'

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>EMH response</th>
<th>MSGQ response</th>
<th>Average EMH response</th>
<th>Average MSGQ response</th>
<th>Program CPU time</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>0.00</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>0.30</td>
<td>0.45</td>
<td>10.878</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CIT801</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>CSC101</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>CSC201</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>CSC5A01</td>
<td>0</td>
<td>207</td>
<td>0.30</td>
<td>0.93</td>
<td>11.070</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CSC5B01</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>CSE0101</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>

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 199.
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

Figure 45. Example of IMS Message Queue Pool Detail by Date tabular report

The report contains this information:

**IMS check point**

The numeric ID of the checkpoint for the IMS session.

**Time**

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

**High QBLK DRRN**

The DRRN of the highest queue block.

**High short message DRRN**

The DRRN of the highest short message queue.

**High long message DRRN**

The DRRN of the highest long message queue.

**Message enqueues.**

The number of calls to enqueue messages.

**Message dequeues.**

The number of calls to dequeue messages.
Queue manager calls
   The total number of calls to QMGR.

Buffer waits
   The number of waits for a free buffer.

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

Ilog waits
   The number of waits for ILOG.

Purge waits
   The number of waits for purge completion.
IMS OSAM/ISAM Buffer Pool Detail by Date report

This report shows the utilization of IMS OSAM/ISAM buffers and pools for the day and time period selected. Values are shown cumulatively at each IMS checkpoint. Figure 46 shows an example of a report.

This information identifies the report:

- **Report ID**: IMSS02
- **Report group**: IMS reports
- **Source**: IMS_CHKPT_IOSAM_T
- **Attributes**: IMS, buffers, utilization, performance, daily, OSAM, ISAM
- **Variables**: MVS_SYSTEM_ID, IMS_SYSTEM_ID, DATE, FROM_TIME, TO_TIME

<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 02.16.11</td>
<td>0</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 02.18.22</td>
<td>8953</td>
<td>8550</td>
<td>142</td>
<td>0</td>
<td>0</td>
<td>108</td>
<td>0</td>
</tr>
<tr>
<td>ALL</td>
<td>3 02.20.17</td>
<td>19403</td>
<td>18720</td>
<td>316</td>
<td>0</td>
<td>0</td>
<td>243</td>
<td>0</td>
</tr>
<tr>
<td>ALL</td>
<td>4 02.45.06</td>
<td>24416</td>
<td>23572</td>
<td>426</td>
<td>0</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.
**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

---

<table>
<thead>
<tr>
<th>Buffer size</th>
<th>IMS check point</th>
<th>Number of buffers</th>
<th>VSAM GET calls</th>
<th>VSAM GET reads</th>
<th>FROM_TIME</th>
<th>TO_TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>139264</td>
<td>1 02.16.11</td>
<td>320</td>
<td>30</td>
<td>295</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>139264</td>
<td>2 02.18.22</td>
<td>320</td>
<td>2558</td>
<td>2439</td>
<td>1800</td>
<td>1319</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>292</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>139264</td>
<td>3 02.20.17</td>
<td>320</td>
<td>6196</td>
<td>5069</td>
<td>4997</td>
<td>2870</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>777</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>160</td>
<td>3</td>
</tr>
<tr>
<td>139264</td>
<td>4 02.45.06</td>
<td>320</td>
<td>8145</td>
<td>6399</td>
<td>6760</td>
<td>3658</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1045</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>214</td>
<td>5</td>
</tr>
<tr>
<td>2048</td>
<td>1 02.16.11</td>
<td>40</td>
<td>207</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2048</td>
<td>2 02.18.22</td>
<td>40</td>
<td>870</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2048</td>
<td>3 02.20.17</td>
<td>40</td>
<td>1668</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2048</td>
<td>4 02.45.06</td>
<td>40</td>
<td>2041</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4096</td>
<td>1 02.16.11</td>
<td>240</td>
<td>72</td>
<td>72</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4096</td>
<td>2 02.18.22</td>
<td>240</td>
<td>2448</td>
<td>3148</td>
<td>1817</td>
<td>1272</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>294</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>61</td>
<td>1</td>
</tr>
<tr>
<td>4096</td>
<td>3 02.20.17</td>
<td>240</td>
<td>5966</td>
<td>5327</td>
<td>4871</td>
<td>2766</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>776</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>160</td>
<td>3</td>
</tr>
<tr>
<td>4096</td>
<td>4 02.45.06</td>
<td>240</td>
<td>7688</td>
<td>4197</td>
<td>6607</td>
<td>3534</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1804</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>214</td>
<td>5</td>
</tr>
<tr>
<td>8192</td>
<td>1 02.16.11</td>
<td>40</td>
<td>8</td>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8192</td>
<td>2 02.18.22</td>
<td>40</td>
<td>110</td>
<td>71</td>
<td>63</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8192</td>
<td>3 02.20.17</td>
<td>40</td>
<td>230</td>
<td>134</td>
<td>126</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8192</td>
<td>4 02.45.06</td>
<td>40</td>
<td>277</td>
<td>161</td>
<td>153</td>
<td>124</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Figure 47. Example of IMS VSAM Buffer Pool Detail by Date tabular report**

The report contains this information:

**Buffer size**

The size of the buffers in the buffer pool.

**IMS check point**

The numeric ID of the checkpoint for the IMS session.

**Time**

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

**Number of buffers**

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

**VSAM GET calls**

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

**VSAM reads**

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

**Retrieves by RBA**

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

**Retrieves by key**

The number of requests to retrieve by key for the IMS session.
VSAM user writes
The number of VSAM user write requests for the IMS session.

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

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

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

<table>
<thead>
<tr>
<th>Region job 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 calls</th>
<th>Fast Path 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>MSGSWMCH</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>IMSR</td>
<td>0</td>
<td>1041</td>
<td>0.00</td>
<td>0.65</td>
<td>60.731</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

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 199.
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

<table>
<thead>
<tr>
<th>Region job name</th>
<th>Tr cl</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</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>MSGSSWCH</td>
<td>0</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>IMSR 1</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 3</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 5</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 2</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 4</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 1</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>
<td>0</td>
</tr>
<tr>
<td>IMSR 2</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>
<td>0</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 199

Tivoli Decision Support for z/OS Report: IMSY06

Appendix A. Reports 209
**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

![IMS Resource Detail by Quarter Hour report](image)

**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.

For descriptions of the other columns in this report, see "IMS User ID Response Time and CPU Detail by Date report" on page 199.
IMS worst case reports

IMS worst case reports are low resolution tabular reports that reflect worst case utilization.

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>DLI database calls</th>
<th>Fast Path DEBB calls</th>
<th>Fast Path MSBB calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSDE0101</td>
<td>0</td>
<td>26</td>
<td>0.00</td>
<td>1.14</td>
<td>1.413</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>1.117</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>0.920</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>0.850</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>0.781</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>0.684</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>0.199</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 199
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

### IMS Program Utilization Worst Case by Date report

**Date**: 2000-02-26  
**System**: 'MVS2'  
**IMS System**: 'IMS2'

<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 calls</th>
<th>Fast Path DEDB calls</th>
<th>Fast Path MSOB 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>
<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>
<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>
<td>0</td>
</tr>
<tr>
<td>PPIT6A01</td>
<td>0</td>
<td>175</td>
<td>0.00</td>
<td>1.08</td>
<td>9.075</td>
<td>0</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>
<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>
<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>
<td>0</td>
</tr>
<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>
<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>
<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 199
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

<table>
<thead>
<tr>
<th>Date</th>
<th>Up Hours</th>
<th>Up In 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>
<td>95.00</td>
</tr>
<tr>
<td>2002-09-28</td>
<td>24</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
<tr>
<td>2002-09-29</td>
<td>24</td>
<td>9</td>
<td>100.00</td>
<td>95.00</td>
</tr>
<tr>
<td>2002-09-30</td>
<td>3</td>
<td>3</td>
<td>33.33</td>
<td>95.00</td>
</tr>
</tbody>
</table>

Tivoli Decision Support for z/OS 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.
Reports

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

Objective (%)  
Availability objective for this resource in the scheduled hours.
IMS CSQ Region Availability, Daily Overview report

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

This information identifies the report:

**Report ID**
CSQA02

**Report group**
CSQ reports

**Source**
IMS_AVAILABILITY_D

**Attributes**
IMS, Availability, Daily, Overview

**Variables**
IMS_System_ID, Date

### IMS CSQ Region Availability, Daily Overview

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

Tivoli Decision Support for z/OS Report: CSQA02

*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_AVAILRESOURCE 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_AVAILRESOURCE 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

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

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

ShMsg
Average Full Function Short messages on queue before processing.

LgMsg
Average Full Function Long messages on queue before processing.

Processed
Total number of messages processed.

AvgTime
Average time (in seconds) each message remains on queue.
IMS 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**: 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 219

---

<table>
<thead>
<tr>
<th>Time</th>
<th>Transaction Name</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>15.15</td>
<td>HC02HR0</td>
<td>-</td>
<td>1.00</td>
<td>0.00</td>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02MR0</td>
<td>-</td>
<td>1.00</td>
<td>0.00</td>
<td>2</td>
<td>0.40</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02MR0</td>
<td>-</td>
<td>0.00</td>
<td>4.00</td>
<td>2</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M100T</td>
<td>-</td>
<td>0.00</td>
<td>1.00</td>
<td>2</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M100T</td>
<td>-</td>
<td>0.00</td>
<td>3.50</td>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M100T</td>
<td>-</td>
<td>0.00</td>
<td>8.00</td>
<td>2</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M110T</td>
<td>-</td>
<td>0.00</td>
<td>11.00</td>
<td>3</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M100T</td>
<td>-</td>
<td>0.00</td>
<td>10.01</td>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>5</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M110T</td>
<td>-</td>
<td>0.00</td>
<td>13.25</td>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M110T</td>
<td>-</td>
<td>0.00</td>
<td>5.00</td>
<td>2</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>HC02M110T</td>
<td>-</td>
<td>0.00</td>
<td>9.00</td>
<td>1</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Figure 57. Example of IMS Message Queue Utilization by Transaction, Date report

Tivoli Decision Support for z/OS Report: CSQTQ02
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
Source IMS_TRAN_QUEUE_DV
Attributes IMS, Transaction, Queue, Date
Variables Origin_IMS, From_Date, To_Date

IMS Msg Queue Utilization Overview, Daily Report
Date: '2002-07-17' to '2002-07-17'
IMS System: 'CSSD'

<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>MSGQ LOCAL</td>
<td>-</td>
<td>0.24</td>
<td>10.56</td>
<td>3841</td>
<td>0.36</td>
<td>-</td>
<td>5.20</td>
<td>0.00</td>
<td>3609</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

Tivoli Decision Support for z/OS Report: CSQTQ03

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 219.
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 219.
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>CPU Average Utilization</th>
<th>Average Response Time</th>
<th>Input Local Queue time</th>
<th>Output CQS Queue time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A70200</td>
<td>6.83000E+02</td>
<td>3.903E+00</td>
<td>1.878E-01</td>
<td>6.830E+02</td>
</tr>
<tr>
<td>A70210</td>
<td>2.00000E+00</td>
<td>0.000E+00</td>
<td>1.500E-01</td>
<td>2.000E+00</td>
</tr>
<tr>
<td>A7041030</td>
<td>4.20000E+01</td>
<td>4.748E-03</td>
<td>3.333E-02</td>
<td>4.200E+01</td>
</tr>
<tr>
<td>A7042030</td>
<td>4.60000E+01</td>
<td>4.845E-03</td>
<td>3.804E-01</td>
<td>4.600E+01</td>
</tr>
<tr>
<td>A7051030</td>
<td>1.36000E+02</td>
<td>1.299E-02</td>
<td>1.390E-01</td>
<td>1.360E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Average</th>
<th>Average</th>
<th>Average</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Time</td>
<td>Process Time</td>
<td>Local Queue Time</td>
<td>Output CQS Queue Time</td>
</tr>
<tr>
<td>1.025E-03</td>
<td>1.114E-01</td>
<td>6.630E+02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>2.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>3.333E-02</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>2.913E-01</td>
<td>4.600E+01</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>0.000E+00</td>
<td>6.912E-02</td>
<td>1.340E+02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>Total Network Responses Time</td>
<td>6.630E+02</td>
<td>7.738E-02</td>
<td></td>
</tr>
<tr>
<td>2.000E+00</td>
<td>1.500E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.600E+01</td>
<td>8.913E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.340E+02</td>
<td>7.090E-02</td>
<td></td>
<td></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 z/OS with QMF (QMFUSE=YES coded in the Tivoli Decision Support for z/OS initialization member DRLFPROF). Figure 61 on page 228 shows an example of a report.

This information identifies the report:

- Report ID: CSQTQ06
- Report group: CSQ reports
- Source: IMS_TRAN_D
- Attributes: IMS, Lterm, Utilization, Performance, Daily
- Variables: Origin_IMS, Process_IMS, Date, origin_Lterm, User_ID
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.

---

**Figure 61. Example of IMS CSQ Transit Time Analysis By LTERM and Userid, Daily report.**

The report contains this information:
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.
Reports

**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 z/OS with QMF (QMFUSE=YES coded in the Tivoli Decision Support for z/OS 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:
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  
The 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 235 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
The report contains this information:

**Date**  
The date of the day for the measurement.

**IMS System ID**  
The name of the IMS subsystem effectively processing the transaction.

**Transaction name**  
The name of the IMS transaction.

**Total Transactions**  
Total number of transactions processed by a PSB.

**Program CPU time**  
The total dependent region CPU seconds.

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

---

### IMS CSQ Resource Utilization, Daily Overview

**IMS SUBSYSTEM NAME:'IMSE'**

**DATE:** 2003-03-11

<table>
<thead>
<tr>
<th>Transaction name</th>
<th>Total CPU time</th>
<th>DC calls</th>
<th>DC calls</th>
<th>DC calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BMP</td>
<td>2.00521E-01</td>
<td>0.000E+00</td>
<td>0.000E+00</td>
<td>1.000E+00</td>
</tr>
<tr>
<td>AUSWAHL</td>
<td>1.02109E+00</td>
<td>0.000E+00</td>
<td>4.440E+02</td>
<td>4.440E+02</td>
</tr>
<tr>
<td>A5760010</td>
<td>1.51792E+00</td>
<td>0.000E+00</td>
<td>1.440E+02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>A5770010</td>
<td>6.69531E-02</td>
<td>0.000E+00</td>
<td>6.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>A5780010</td>
<td>2.27839E-01</td>
<td>0.000E+00</td>
<td>3.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>A5790010</td>
<td>5.07031E-02</td>
<td>0.000E+00</td>
<td>2.000E+00</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>A5920010</td>
<td>3.31427E+00</td>
<td>0.000E+00</td>
<td>3.320E+02</td>
<td>0.000E+00</td>
</tr>
<tr>
<td>A70200</td>
<td>3.90340E+00</td>
<td>0.000E+00</td>
<td>6.830E+02</td>
<td>6.830E+02</td>
</tr>
</tbody>
</table>

### DB calls

<table>
<thead>
<tr>
<th>DC</th>
<th>GHN</th>
<th>GHNP</th>
<th>GHU</th>
<th>GN</th>
<th>GNP</th>
</tr>
</thead>
<tbody>
<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>

### DB calls

<table>
<thead>
<tr>
<th>GU</th>
<th>ISRT</th>
<th>REPL</th>
</tr>
</thead>
<tbody>
<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>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**  
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 238 shows an example of a job that you can use to assemble the log records mapping macro ILOGREC that is provided with IMS. The sample shown is the most inclusive method of extracting DSECTs. You can modify it to suit your own needs. You can write your own record procedures that use the composite record created by the IMS Performance feature.
Creating IMS log record DSECTs

//USERIDA JOB (ACC000,001), 'ILOGREC',
// NOTIFY=USERID, MSGCLASS=A, CLASS=A, REGION=0M
//DSECT PROC
//HASM EXEC PGM=IEV90, PARM='NODECK, NOXREF, LIST, NORD, NOOBJECT'
//SYSLIB DD DISP=SHR, DSN=IMS71.GENLIBA, DCB=SYS1.MACLIB
// DD DISP=SHR, DSN=IMS71.GENLIB
// DD DISP=SHR, DSN=SYS1.MACLIB
// SYSLIB DD DISP=SHR, DSN=IMS71.GENLIB
// SYSLIB DD DISP=SHR, DSN=SYS1.MACLIB
// SYSLIB DD DISP=SHR, DSN=SYS1.MACLIB
// SYSLIB DD DISP=SHR, DSN=SYS1.MACLIB
// SYSLIB DD DISP=SHR, DSN=SYS1.MACLIB
// SYSUT1 DD UNIT=SYSDA, SPACE=(CYL,(2,1))
// SYSUT2 DD UNIT=SYSDA, SPACE=(CYL,(2,1))
// SYSUT3 DD UNIT=SYSDA, SPACE=(CYL,(2,1))
// SYSPRINT DD SYSOUT=*  
// PEND
// D1 EXEC DSECT /* Full function record DSECTs */
// SYSin DD *
// ILOGREC RECID=ALL
// END
/*
// D2 EXEC DSECT /* Fast path record DSECTs */
// SYSin DD *
// DBFLGGRIM
// EJECT
// DBFLGROM EJECT
// DBFBMSDB EJECT
// DBFDQLC EJECT
// DBFLGRDQ EJECT
// DBFLGSYN EJECT
// DBFLGRIC EJECT
// DBFLSRRT RECID=50 EJECT
// DBFLGRSD EJECT
// DBFLGRR EJECT
// END
/*
// D3 EXEC DSECT /* IMS control block DSECTs */
// SYSin DD *
// IBPOOL EJECT
// IDLIVSAM BFSP EJECT
// ICL1 CTTBASE=1 EJECT
// ICL1 CNTBASE=1 EJECT
// ICL1 SPQBASE=1 EJECT
// ICL1 CIBBASE=1 EJECT
// ICL1 CIBBASE=1 EJECT

Figure 64. Sample JCL for assembling IMS log record DSECTs (Part 1 of 3)
EJECT
ICLI PCIBASE=1
EJECT
ICLI CRBBASE=1
EJECT
ICLI CLBBASE=1
EJECT
ICLI CVBBASE=1
EJECT
ICLI CBBBASE=1
EJECT
ICLI BUFBASE=1
EJECT
ICLI POOBASE=1
EJECT
ICLI BFRBASE=1
EJECT
ICLI QDSBASE=1
EJECT
ICLI QPCBASE=1
EJECT
ICLI QLGBASE=1
EJECT
ICLI FREBASE=1
EJECT
ICLI FBPBASE=1
EJECT
ICLI SPABASE=1
EJECT
ICLI FEIBBASE=1
EJECT
IAPS SMMBASE=1
EJECT
IAPS TCTBASE=1
EJECT
IDL I DBSBASE=1
EJECT
IDL I DDRBASE=1
EJECT
IDL I DMBBASE=1
EJECT
IDL I DPCBASE=1
EJECT
IDL I DSGBASE=1
EJECT
IDL I FDBBASE=1
EJECT
IDL I FLDBASE=1
EJECT

Figure 64. Sample JCL for assembling IMS log record DSECTs (Part 2 of 3)
IDLI  FSBBASE=1
EJECT
IDLI  FUNBASE=1
EJECT
IDLI  JCBBASE=1
EJECT
IDLI  LEVBASE=1
EJECT
IDLI  PDRBASE=1
EJECT
IDLI  PDVBASE=1
EJECT
IDLI  PSBBASE=1
EJECT
IDLI  PSBBASE=1
EJECT
IDLI  PSPBASE=1
EJECT
IDLI  PSTBASE=1
EJECT
IDLI  PS24BASE=1
EJECT
IDLI  SDBBASE=1
EJECT
IDLI  SMBASE=1
EJECT
IDLI  TPCBASE=1
EJECT
IDLI  CALLER=1
EJECT
ISCD
END

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 242] 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 z/OS CNTL library.

For information about writing Tivoli Decision Support for z/OS record procedures, refer to the Language Guide and Reference.
MACRO
DRLJXIDC &PFX=BASIC,&TYPE=DSECT
AIF ('&TYPE' EQ 'DSECT' OR '&TYPE' EQ 'O').NEXT1
&RTYPE SETC 'EQU'
&O SETC '.'
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').LFSP
AIF ('&PFX' EQ 'ESS').LESS
AIF ('&PFX' EQ 'EMH').LEMH
MNOTE &O,'INVALID PREFIX TYPE SPECIFIED'
MEXIT
LBASIC ANOP
CMREC &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
CMF1WFI EQU X'80' TXN IS WFI
CMF1AOI EQU X'40' TXN STARTED BY AOI EXIT
CMF1PMNT EQU X'20' PARENT TRANSACTION
CMF1ROOT EQU X'10' ROOT TRANSACTION
CMF1PSCH EQU X'08' PSEUDO SCHEDULE
CMF1PEND EQU X'04' PSEUDO END
CMF1ABND EQU X'02' TRANSACTION ABENDED
CMFIREQW EQU X'01' TRANSACTION RE-ENQUEUED
CMFLAG2 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
CMPEDATE DS PL4 DATE STAMP OF PSEUDO END
CMPETIME DS PL4 TIME STAMP OF PSEUDO END
CMPEQTM EQU 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)
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
CMD1NOSEG DS H NUMBER OF INPUT SEGMENTS
CMD1OUTS DS H NUMBER OF ACTUAL OUTPUTS
MEXIT
LD2 ANOP
CMD2SECT &RTYPE &O
*****************************************************************
* PREFIX OF THE D2 SECTION
* *****************************************************************
CMCID 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
CM2X03 EQU X'80' X03 RECORD(S) PRESENT
CM2X35 EQU X'40' X35 RECORD(S) PRESENT
CM2X31 EQU X'20' X31 RECORD PRESENT
CM2X36 EQU X'10' X36 RECORD(S) PRESENT
CM2P5SW EQU X'08' PROGRAM SWITCH (DEST IS SMB)
CM2MCAN 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 &O

Figure 65. DRLJXIDC MACRO (Part 3 of 5)
Figure 65. DRLJXIDC MACRO (Part 4 of 5)
* 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
CMEMHSEC &RTYPE &D
MEXIT
LEMH ANOP
*****************************************************************************

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 z/OS CNTL library. Figure 66 on page 248 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 z/OS
* /IMS
* COPYRIGHT: NONE
* STATUS: IMS/ESA V3R1
* FUNCTION:
* WRITES THE RECORDS USED BY Tivoli Decision Support for z/OS
* /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 z/OS/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
Appendix E. DFSLTMG0 log merge utility

The DFSLTMG0 log merge utility produces one data set by merging the system log data sets (SLDS) from two or more IMS systems. The log mMerge utility can merge up to nine IMS system logs from the same IMS release. Each log is the output of a uniquely identified IMS system running during the same time span. The order of input to the log merge utility is LOG01, LOG02... LOG09. DFSLTMG0 is placed in IMS.RESLIB during IMS system definition.

```
//STEP0 EXEC PGM=DFSLTMG0
//STEPLIB DD DSN=IMS710.RESLIB, DISP=SHR
//SYSPRINT DD SYSOUT=*  
//PRINT DD SYSOUT=*  
//LOG01 DD DSN=xxx.IMSLOG01, DISP=OLD  
//LOG02 DD DSN=xxx.IMSLOG02, DISP=OLD  
//LOGOUT DD DSN=xxx.IMSLOG.MERGE, DISP=(NEW,CATLG)  
//       VOL=SER=yyyy, UNIT=3390,SPACE=(CYL, (100,5)), 
//       DCB=(RECFM=VB, LRECL=27994, BLKSIZE=32760)  
//SYSIN DD *  
START 01157,0001  
STOP 01157,2359  
MSG  
//*
```

Figure 67. DFSLTMG0 log merge utility

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 DSN=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 DSN=IMS.LOGA,DISP=OLD,  
// VOL=SER=XXXXXX, UNIT=TAPE
```

**LOG02 DD**

Describes the second input log data set.

```
//LOG02 DD DSN=IMS.LOGB,DISP=OLD,  
// VOL=SER=XXXXXX, UNIT=TAPE
```

**LOGOUT DD**

Describes the output data set.

```
//LOGOUT DD DSN=IMS.LOGOUT,DISP=(,PASS),  
// VOL=SER=YYYYY,UNIT=TAPE,  
// DCB=(RECFM=V8, LRECL=6000, BLKSIZE=6008)
```

**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, hhmmsstt).
   START

STOP
   You must specify a stop time, which must be relative to the time field in
   LOG01 (yyddd, hhmmsstt)
   STOP

Log Record Selection
   Use this control statement to merge only certain types of log records.

Controlling the log merge

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
DEDDB 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
RECAN  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>Abbreviation</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
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM’s suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to
IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM’s application programming interfaces.

If you are viewing this information in softcopy form, the photographs and color illustrations might not display.

**Trademarks**

AS/400, CICS, DB2, DB2 Universal Database, eServer, GDDM, IBM, the IBM logo, IMS, IMS/ESA, iSeries, Lotus, MVS, NetView, OS/390, Passport Advantage, pSeries, QMF, RACF, RMF, Rational, Redbook, Tivoli, Tivoli Enterprise, Tivoli Enterprise Console, VTAM, WebSphere, z/OS, and zSeries are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds 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.

Other company, product, and service names may be trademarks or service marks of others.
Glossary

A
administration. A Tivoli Decision Support for z/OS task that includes maintaining the database, updating environment information, and ensuring the accuracy of data collected.

administration dialog. A set of host windows used to administer Tivoli Decision Support for z/OS.

C
collect. A process used by Tivoli Decision Support for z/OS to read data from input log data sets, interpret records in the data set, and store the data in DB2 tables in the Tivoli Decision Support for z/OS database.

component. An optionally installable part of a Tivoli Decision Support for z/OS feature.

control table. A predefined Tivoli Decision Support for z/OS table that controls results returned by some log collector functions.

D
data table. A Tivoli Decision Support for z/OS 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 z/OS 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 z/OS program that processes log data sets and provides other Tivoli Decision Support for z/OS services.

log collector language. Tivoli Decision Support for z/OS statements used to supply definitions to and invoke services of the log collector.

log data set. Any sequential data set that is used as input to Tivoli Decision Support for z/OS.

log definition. The description of a log data set processed by the log collector.

log procedure. A program module that is used to process all record types in certain log data sets.

lookup expression. An expression that specifies how a value is obtained from a lookup table.

lookup table. A Tivoli Decision Support for z/OS DB2 table that contains grouping, translation, or substitution information.

P
purge condition. Instruction for purging old data from the Tivoli Decision Support for z/OS database.

R
record definition. The description of a record type contained in the log data sets used by Tivoli Decision Support for z/OS, including detailed record layout and data formats.

record procedure. A program module that is called to process some types of log records.

record type. The classification of records in a log data set.

repeated section. A section of a record that occurs more than once, with each occurrence adjacent to the previous one.

report definition language. Tivoli Decision Support for z/OS statements used to define reports and report groups.

report group. A collection of Tivoli Decision Support for z/OS reports that can be referred to by a single name.
**reporting dialog.** A set of host or workstation windows used to request reports.

**resource group.** A collection of network resources that are identified as belonging to a particular department or division. Resources are organized into groups to reflect the structure of an organization.

**resource information.** Environment information that describes the elements in a network.

**section.** A structure within a record that contains one or more fields and may contain other sections.

**source.** In an update definition, the record or DB2 table that contains the data used to update a Tivoli Decision Support for z/OS DB2 table.

**system table.** A DB2 table that stores information that controls log collector processing, TTivoli Decision Support for z/OS dialogs, and reporting.

**target.** In an update definition, the DB2 table in which Tivoli Decision Support for z/OS stores data from the source record or table.

**threshold.** The maximum or minimum acceptable level of usage. Usage measurements are compared with threshold levels.

**Tivoli Decision Support for z/OS database.** A set of DB2 tables that includes data tables, lookup tables, system tables, and control tables.

**update definition.** Instructions for entering data into DB2 tables from records of different types or from other DB2 tables.

**view.** An alternative representation of data from one or more tables. A view can include all or some of the columns contained in the table on which it is defined.
Index

A
abend codes from transaction 45
abend during collect commit 45
accessibility xiv
active region statistics record definition 58
additional available IMS record information 45
administering the IMS Performance feature 31
AOI user exit initiated transaction special log procedure case 16, 115
APPC_MODE_NAME data column 73
APPC_NAME data column 73
APPC_NETID data column 72, 73
APPC_TIBQAB_A data column 73
APPC_TIBQAB_T data column 73
APPC_TPI_COMMITS data column 72
APPC_TPI_CPU_SEC data column 72
APPC_TPI_DPSBCALLS data column 72
APPC_TPI_ENDED data column 72
APPC_TPI_STARTED data column 72
application reports
  CPU utilization trend graphical 192
  response time overview graphical 185, 213, 215, 216, 219, 221, 222, 223, 224, 227, 231
  response time trend graphical 189
  transaction overview graphical 187
  transaction trend graphical 191
Application subcomponent tables 87
APPLICATION_NAME data column 100
APPLICATION_NAME key column 70, 87, 88
archive exit for IMS 247
attributes of reports 178
audience, intended  xi
BUFFERS data column 90, 98
BUFFERS_CURRENT data column 91
BUFFERS_HIGH data column 91
BUFFERS_LOCKED data column 90
BUFFERS_OVERSIZE data column 91
BUFFERS_SEARCHED data column 90
C
chart report 179
checkpoint record definitions
  batch 57
  begin 54
  CCB 55
  CIB 55
  CLB 55
  CNT 54
  CTB 54
  DMB 54
  end blocks 56
  end queues 57
  EQE 55
  Fast Path ADSC 56
  Fast Path begin 56
  Fast Path buffer 56
  Fast Path DMHR/DEDDB 56
  Fast Path ECNT 56
  Fast Path EMHB 56
  Fast Path end 56
  Fast Path IEEQE 56
  Fast Path RCTE 56
  message queue 54
  MSDB begin 55
  MSDB ECNT 56
  MSDB end 56
  MSDB header 56
  MSDB pageable 56
  MSDB pagefixed 56
  PSB 54
  SMB 54
  SPA 55
  SPA QB 55
  VTCB 55
CHKPT_ADDR_SPC_ID data column 94
CHKPT_CPU_ID data column 94
CHKPT_CTRL_TCB data column 94
CHKPT_DLI_TCB data column 94
CHKPT_HOT_STANDBY data column 94
CHKPT_IRLM_NAME data column 94
CHKPT_LMSG_RECORDS data column 95
CHKPT_QBLK_RECORDS data column 95
CHKPT_SMMSG_RECORDS data column 95
CI_FOUND_IN_POOL data column 98
CICSNAME parameter 31, 33
collect commit abend 45
collect for large installation 42
command log record definition 49
common reference columns 67
communications trace record definition 60
collection code from transaction 45
component
customizing the IMS Performance feature  6, 108
control tables
control tables
conversational transaction special log procedure case  16, 116
CPU and userid response time by date detail report  199
CPU utilization report
application trend graphical  192
system trend graphical  197
customer support xiv
d day flow  23
through DRL2LOGP  27, 121
through log collector  25, 120
through the IMS Performance feature  28
through Tivoli Decision Support for z/OS  23
data tables
built by IMS Shared Queue feature  139
descriptions  67
for application subcomponent  68
for application subcomponent tables  87
for IMS collect component  68
for IMS log records component  68
for statistics subcomponent  68, 89
for system subcomponent  68, 85
for transaction subcomponent  68, 71
for transaction transit time component  139
IMS_APPLICATION_H_._W  87
IMS_CHKPT_IOSAM_T  89
IMS_CHKPT_POOLS_T  91
IMS_CHKPT_REGION_T  92
IMS_CHKPT_STATS_T  94
IMS_CHKPT_VSAM_T  98
IMS_PSB_ACCOUNT_H_._D_._W  152
IMS_SYSTEM_Q_._D  85
IMS_SYSTEM_TRAN_H_._D  147
IMS_TRAN_H_._D_._W  140
IMS_TRAN_QUEUE_Q_._D  144
IMS_TRANSACTION_H_._D_._W  71
IMS_USER_APPN_D_._D_._W  88
IMS_USER_TRAN_H_._D_._W  82
name suffix  67
naming standard  67
summarization level  67
data tables built by IMS Performance feature  67
database record definitions
buffer pool statistics  57
close  52
error  52
open  52
DATE key column  68, 70, 71, 82, 85, 87, 88, 89, 91, 92, 94, 98
DAY_OF_WEEK control table  68
DB2 tables
built by IMS Shared Queue feature  139
DB2 tables built by IMS Performance feature  67
ddnames for DRL2LOGP
DRLDUMP  40, 131
DRLCHKI  40, 130
DRLCHKO  40, 130
DRLCOMP  40, 130
DRLIPARM  41, 131
DRLIRPTn  40, 131
DRLOUT  40, 131
defining the operating environment  5, 107
descriptions in tables  67
detail report  178
DSPCBTO0 storage statistics record definition  58
discrete IMS sessions  42
dispatch storage statistics record definition  57
DLI utilization system trend graphical report  198
DLI_CMD_CALLS data column  73
DLI_EXCL_DEQUEUES data column  75
DLI_EXCL_ENQUEUE data column  75
DLI_EXCL_ENQWAITS data column  75
DLI_GCMD_CALLS data column  76
DLI_QCMD_CALLS data column  76
DLI_QCMD_ENQUEUE data column  76
DLI_QCMD_ENQWAITS data column  76
DLI_TEST_DEQUEUES data column  76
DLI_TEST_ENQUEUE data column  76
DLI_TEST_ENQWAITS data column  76
DLI_UPDATE_DEQUEUES data column  76
DLI_UPDATE_ENQUEUE data column  76
DLI_UPDATE_ENQWAITS data column  76
DLIDB_CALLS data column  73
DLIDB_DLET_CALLS data column  74
DLIDB_GHNCALLS data column  74
DLIDB_GHNPCALLS data column  74
DLIDB_GHU_CALLS data column  74
DLIDB_GN_CALLS data column  74
DLIDB_GNP_CALLS data column  74
DLIDB_GU_CALLS data column  74
DLIDB_Isrt_CALLS data column  74
DLIDB_REPL_CALLS data column  75
DLIDBCTL_DBIOS_SEC data column  74
DLIDBCTL_DBIOS data column  73
DLIDBCTL_LOCK_SEC data column  74
DLIDC_GN_CALLS data column  75
DLIDC_GU_CALLS data column  75
DLIDC_Isrt_CALLS data column  75
DLIDC_PURGE_CALLS data column  75
DLISQ6_ACCUM_SEC data column  76
DMHR on I/O error record definition  60
DRL2LOGP
data flow  27, 121
data format  40, 130
ddnames  40, 130
description  40, 130
JCL example  41, 131
output record layout  241
parameters  31
running  41, 131
DRL3nn0 record procedure  20
DRL3nn1 record procedure  21
DRL3nn2 record procedure  21
DRLDUMP ddbname  40, 131
DRLCHKI ddbname  40, 130
DRLCHKO ddbname  40, 130
Fast Path record definition (continued)
dequeue message 59
hot standby MSDB relocation 60
input message 58
MSDB change 59
output message 59
sequential dependent syncpoint 59
Fast Path special log procedure case 17, 116
FASTPATH parameter 32, 34
FBP_DIR_IO_COUNT data column 95
FBP_DIR_NO_ENTRIES data column 95
FBP_DIR_REQUESTS data column 95
FBP_FBLOCK_IFETCHQ data column 95
FBP_FBLOCK_IGNORES data column 95
FBP_FBLOCK_REQ data column 95
FBP_FBLOCK_WASHES data column 95
FBP_IFETCH_FBLOCKQ data column 95
FBP_IFETCH_FBLOCKQ data column 95
FBP_IFETCH_FBLOCKQ data column 95
FBP_FBLOCK_FBLOCKQ data column 95
FBP_FBLOCK_FBLOCKQ data column 95
FBP_FBLOCK_FBLOCKQ data column 95
FBP_FBLOCK_FBLOCKQ data column 95
FF composite record section 63, 138
flow of data through Tivoli Decision Support for z/OS 23
format buffer pool statistics record definition 57
format of reports 177
FF composite record section 64
FP_CI_HNH_CONT data column 78
FP_CI_HNH_CONT data column 78
FP_COMBINATIONS data column 78
FP_COMBINATIONS data column 78
FP_DEDB_BFR_WAITS data column 78
FP_DEDB_CALLS data column 78
FP_DEDB_HSSP_CALLS data column 78
FP_DEDB_HSSP_PUTS data column 78
FP_DEDB_NH_PUTGETS data column 78
FP_LOGGED_CI data column 78
FP_MSDB_CALLS data column 78
FP_OVERFLOW_BFR data column 78
FP_UOW_HNH_CONT data column 79
FP_UOW_NNH_CONT data column 79
front end switching special log procedure case 17, 117
full function database record definition
  update undo/KSDS insert 58
  update undo/redo successful 58
  update unsuccessful 58
FULLFUNC parameter 31, 34

G

glossary 259
graphical report 179
GROUP_ID key column 101
GROUP_ID lookup table 101

H

H summarization level suffix 67
HS_BUFFERS data column 99
ID for report 177
ILOGREC sample JCL 237
IMS
  event accounting record definition 50
  log record DSECTs 237
IMS collect component tables 68
IMS CSQ Resource Utilization, Daily Overview 234
IMS CSQ Utilization Reports
  IMS CSQ Resource Utilization, Daily Overview 234
IMS Light feature, running 133
IMS log records component tables 68
IMS Performance feature
  administration 31
  capability maximization 42
  data flow 28
IMS_APPLICATION lookup table 8, 100
IMS_APPLICATION_H, _W tables 87
IMS_APLID 38, 128
IMS_APLID common reference column 72, 89, 91, 92, 94, 98
IMS_AVAILABILITY common reference column 72, 89, 91, 92, 94, 98
IMS_AVAILABILITY_D, W lookup table 159
IMS_AVAILABILITY_T lookup table 158
IMS_CHECKPOINT common reference column 89, 91, 92, 94, 98
IMS_CHKPT_IOSAM_T table 89
IMS_CHKPT_POOLS_T table 91
IMS_CHKPT_REGION_T table 92
IMS_CHKPT_STATS_T table 94
IMS_CHKPT_VSAM_T table 98
IMS_CTRL_REGION 38
IMS_CTRL_REGION common reference column 72, 89, 91, 92, 94, 98
IMS_PSB_ACCOUNT_H, D, W data tables 152
IMS_SYSTEM_ID 38, 128
IMS_SYSTEM_ID key column 70, 71, 82, 85, 87, 88, 89, 91, 92, 94, 98, 100
IMS_SYSTEM_Q, D tables 85
IMS_SYSTEM_TRAN_H, D tables 147
IMS_TRAN_H, D, W tables 140
IMS_TRAN_QUEUE_Q, D tables 144
IMS_TRAN_QUEUE_QV, DV views 151
IMS_TRAN_TYPE key column 70, 83
IMS_TRANSACTION_H, D, _W tables 71
IMS_USER_APLID_H, W tables 88
IMS_USER_TRAN_H, D, _W tables 82
IMS VERSION common reference column 72, 90, 91, 93, 94, 98
IMS_Vnnn_COLLECT log definition 47
IMS_Vnnn_R2 composite record 61
IMS_Vnnn_SLDs log definition 47
IMS_Vnnn_Stxxxx composite record 61
IMS_Vnnn_TRAN composite record 61
IMSID parameter 31, 34, 126, 127
MSIDCHECK parameter 32, 35, 42, 126
MSVID parameter 31, 34, 127
incomplete transaction in composite record 14
input message reenqueue special log procedure case 16
input set (DI) record group 11
installation
  component 7, 109

installation (continued)
  testing 9, 111
  the IMS Performance feature 6, 108
introduction to the IMS Performance feature 3, 105
ISAM/OSAM buffer pool detail by date 204
ISC root information 45
ISC special log procedure case 17, 117

JCL samples
  DRL2LOGP 41, 131
  ILOGREC 237
  log collector 39

K
key column cross-reference 70
key columns
  TRANS_TYPE 149

large installation collect 42
latch statistics record definition 57
LOC_CALLS_WAIT_ID data column 90
LOC_CALLS_WAIT_RD data column 90
LOC_CALLS_WAIT_WR data column 90
LOCATE_CALLS data column 90
LOCATE_CALLS_SUCC data column 90
log
  from different sessions 42
  from different systems 42
  with missing data 42
log collector
  data flow 25, 120
  parameters 38
  running 39, 129
log definition 47
log procedure
  composite records and subtypes 13, 114
  description 11, 113
  DRLOUT reports 18, 117
  parameters 31
  record grouping 11
  release dependency 18
  set relationships 13, 113
  special cases 14, 114
log record definition
  buffer control 57
  data set control 57
log record DSECTs 237
logger statistics record definition 57
logical set of record types 11
LOGICAL_TERMINAL key column 70, 82, 101
LOGL-awesome_WRITE data column 95
LOGL_CHKW_REQUESTS data column 95
LOGL_CURR_SEQ_NO data column 95
LOGL_WTBF_CHKPT data column 95
LOGL_WTBF_NOT_CHKPT data column 95
LOGL_WTWT_REQUESTS data column 95
LookupAt message retrieval tool xiii
lookup table
  definition 8
  GROUP_ID description 101
  IMS application description 100
lookup table (continued)
  updating IMS_APPLICATION  8
  updating non-IMS Performance feature  9, 110
lookup tables
  IMS_AVAILRESOURCE  160
  IMS_AVAILABILITY_D,W  159
  IMS_AVAILABILITY_T  158
  LRECL_ALTERED data column  99

M
M summarization level suffix  67
main pools statistics record definition  57
manuals
  feedback  xii
  online  xii
  ordering  xii
MAXFREE parameter  32, 35, 126, 127
maximizing capabilities of the IMS Performance feature  42
MAXOUTPUT parameter  32, 35, 42
MAXUOR parameter  32, 35, 42
message
  root information  45
message queue pool detail by date report  202
message queue record definition
  cancel  53
  dequeue  53
  DRRN free  53
  enqueue  53
  from CNT  49
  from PSB or IMS  49
  GU  53
  prefix changed  52
  reject  53
  syncpoint fail  54
  syncpoint transfer  54
message retrieval tool, LookAt  xiii
message-driven BMP program special log procedure case  16, 115
multiple outputs special log procedure case  15, 114
multiple segment input special log procedure case  14, 114
multiple segment output special log procedure case  15, 114
multiple system coupling special log procedure case  17, 117
multiple transactions per schedule of PSB special log procedure case  15, 114
MVS_SYSTEM_ID key column  70, 71, 82, 85, 87, 88, 89, 91,
  92, 94, 98, 100
MVS_SYSTEM_ID parameter  38, 128

N
naming standard for tables  67
NEW_BLOCK_CREATES data column  90
non-message-driven BMP program special log procedure case  16, 115

O
OLDS
  padding record  58
  online publications  xiv
operating environment  5, 107
ordering publications  xiv
OSAM/ISAM buffer pool detail by date report  204
output message reenqueue special log procedure case  16, 115
output set (D2) record group  11
overview of key columns  70
overview of Tivoli Decision Support for z/OS data flow  23
overview report  178

P
parameter report from log procedure  18, 117
parameters
  CICSNAME  31, 33
  DRL2LOGP  31
  FASTPATH  32, 34
  FULLFUNC  31, 34
  IMS_APPLICATION  38, 128
  IMS_CTRL_REGION  38
  IMS_SYSTEM_ID  38, 128
  IMSID  31, 34, 126, 127
  IMSIDCHECK  32, 35, 42, 126
  IMSiever  31, 33, 127
  log collector  38
  log procedure  31
  MAXFREE  32, 35, 126, 127
  MAXOUTPUT  32, 35, 42
  MAXUOR  32, 35, 42
  MTOTRAFFIC  32, 35
  MVS_SYSTEM_ID  38, 128
  PASSLOGREC  32, 35
  PSEUDOWFICHK  32, 35
  RECTYPE  32, 35, 127
  REPORTS  31, 33, 126, 127
  RESOLUTION  33, 36
  SQNLOGS  126, 127, 133
  START  32, 36, 126, 127
  STOP  33, 36, 126, 127
  TABLEFLUSH  33, 37, 42, 126, 128
  WRITEPENDING  33, 37, 126, 128
  XRFNAME  33, 37, 126, 128
  XRFSYS  33, 37, 126, 128
  PASSLOGREC parameter  32, 35
pending node report from log procedure  18, 118
QP_QMGR_CALLS data column 90
QP_QMGR_LOCATES data column 96
QP_QMGR_RELEASES data column 96
QP_READ_REQUESTS data column 96
QP_TRANSLATE_REQ data column 96
QP_WAIT_REQUESTS data column 96
QP_WRITE_REQUESTS data column 96
queue pool statistics record definition 57
quick reschedule special log procedure case 17, 116

R
R0 record procedure 20
R1 record procedure 21
R2 record procedure 21
READ_IO_COUNT data column 90
RECANY buffer statistics record definition 58
RECANY_MAX data column 96
RECANY_USED data column 96
record definition 48
record grouping 11
record procedure 19
record sections in composite record 61
record types and subtypes for composite records 66
RECTYPE parameter 32, 35, 127
region and transaction detail by date report 209
region utilization detail by date report 208
REGION_JOB_NAME key column 70, 85, 92
REGION_PST_ID key column 70, 85, 92
report
attribute 178
format 179
ID 177
procedure 19
title 177
type 178
REPORTS parameter 31, 33, 126, 127
RESOLUTION parameter 33, 36
resource report
detail by quarter hour 210
user ID worst case by date 211
response time report
application overview graphical 185, 213, 215, 216, 219,
221, 222, 223, 224, 227, 231
application trend graphical 189
system trend graphical 193
userid and CPU detail by date tabular 199
RESPONSE_BNDY1_SEC data column 81
RESPONSE_BNDY2_SEC data column 81
RESPONSE_BNDY3_SEC data column 81
RESPONSE_BNDY4_SEC data column 81
RETRIEVES_BY_KEY data column 99
RETRIEVES_BY_RBA data column 99
ROOT composite record section 62

S
SCH_BMP_ACTIVE data column 96
SCH_MPP_ACTIVE data column 96
SCH_SMB_INT_CONFL data column 96
SCH_SMB_LOOKED_AT data column 96
SCH_SMB_OTHER data column 97
SCH_SMB_PGM_CONF data column 97
SCH_SMB_PRIOCUTOFF data column 97
scheduling statistics record definition 57
sections of composite record 61

Q
Q summarization level suffix 67
QP_BUFFER_REPOSN data column 96
QP_BUFFER_WAITS data column 96
QP_BUFFERS_LOCKED data column 96
QP_BUFFERS_UNLOCK data column 96
QP_DECBRD_WAIT data column 96
QP_DECWRWRITE data column 96
QP_DDRN_HIGH_LMSG data column 96
QP_DDRN_HIGH_QBLK data column 96
QP_DDRN_HIGH_SMGS data column 96
QP_ENQDEQ_BFRWAIT data column 96
QP_ILOG_WAIT data column 96
QP_IO_ERROR_NORET data column 96
QP_MSG_CANCELS data column 96
QP_MSG_DEQUEUES data column 96
QP_MSG_ENQUEUES data column 96
QP_PCB_UNCHAINS data column 96
QP_PURGE_REQUESTS data column 96
QP_PURGE_WAITS data column 96
QP_PURGE_WRITES data column 96
QP_QMGR_CALLS data column 96

security violation record definition 51
selecting components and subcomponents 7, 109
set relationships 13, 113
shared queue
data tables
IMS_PSB_ACCOUNT_H_D_W 152
IMS_SYSTEM_TRAN_H_D 147
IMS_TRAN_H_D_W 140
IMS_TRAN_QUEUE_Q_D 144
lookup tables
IMS_AVAIL_Resource 160
IMS_AVAILABILITY_D_W lookup table 159
IMS_AVAILABILITY_T 158
views
IMS_TRAN_QUEUE_QV_DV 151
Shared queue
reports samples 180
Shared Queue
Logs 125
Shared queue feature
logs procedure 113
sign-on/off record definition 52
snap trace record definition 60
software support xiv
SPA
composite record section 62
insert record definition 52
SQNLOGS parameter 126, 127, 133
standard report format 179
start of conversation record definition 51
START parameter 32, 36, 126, 127
Statistics subcomponent tables 89
STOP parameter 33, 36, 126, 127
subcomponent selection 7, 109
suffixes in table names 67
summarization level suffixes 67
SYNC_CALLS data column 99
system generated output special log procedure case 16, 115
system report
CPU utilization trend graphical 197
CPU utilization trend tabular 197
dli utilization trend graphical 198
dli utilization trend tabular 198
response time trend graphical 193
response time trend tabular 193
transaction volumes trend graphical 195
transaction volumes trend tabular 195
System subcomponent tables 85

T
T summarization level suffix 67
table descriptions 67
table name suffixes 67
table naming standard 67
TABLEFLUSH parameter 33, 37, 42, 126, 128
tables
built by IMS Shared Queue feature 139
tables built by IMS Performance feature 67
tabular report 179
terminal message switch special log procedure case 16, 115
terms defined 259
testing the installation of the IMS Performance feature 9, 111
time key column 68, 70, 71, 82, 85, 87, 89, 91, 92, 94, 98
timeStamp key column 89, 91, 92, 94, 98
title of report 177
trace table log record definition 60
TRANS_TYPE key column 149
transaction abend/completion codes 45
transaction incomplete in composite record 14
transaction report
application overview graphical 187
application overview tabular 187
application trend graphical 191
application trend tabular 191
region detail by date tabular 209
utilization detail by date tabular 201
volumes for system trend graphical 195
volumes for system trend tabular 195
Transaction subcomponent tables 71
transaction type key column description 83
TRANSACTION_CLASS data column 72
TRANSACTION_CLASS key column 70, 71, 85
TRANSACTION_NAME data column 93
TRANSACTION_NAME key column 70, 71, 82, 100
TRANSACTION_TYPE data column 72
TRANSACTION_TYPE key column 72, 86
trend report definition 178
type of report 178

U
unit-of-recovery set record group 12
updating lookup tables
definition 8
IMS_APPLICATION 8
non-IMS Performance feature lookup tables 9, 110
user ID report
resource worst case by date 211
response time and CPU detail by date 199
USER_ID key column 70, 82, 88, 101
utilization report
program worst case by date 212
region by date detail 208
transaction by date detail 201

V
views
for transaction transit time component 139
IMS_TRAN_QUEUE_QV_DV 151
VSAM buffer pool detail by date report 206
VSAM subpool statistics record definition 57
VSAM_ESDS_INSERTS data column 99
VSAM_GET_CALLS data column 99
VSAM_KSDS_INSERTS data column 99
VSAM_USER_WRITE data column 99
VSAM_READS data column 99
VSAM_SCHBFR_CALLS data column 99
VSAM_USER_WRITE data column 99

W
W summarization level suffix 67
wait for input program special log procedure case 17, 116
WAITS_NO_BUFFER data column 90
WAITS_RLSE_OW data column 90
worst case report 178
WRITEPENDING parameter 33, 37, 126, 128
WRITES_BFR_STEAL data column 90
XRFNAME parameter  33, 37, 126, 128
XRPSYS parameter  33, 37, 126, 128