

Volume

1

ORACLE CORPORATION

**RMU SHOW STATISTIC
DBA HANDBOOK**

(Second Edition)

METHODS AND INTERNALS

**RICK ANDERSON
ORACLE RDB ENGINEERING
REV. 12/15/98**

© COPYRIGHT 1998, ORACLE CORPORATION. ALL RIGHTS RESERVED.

Oracle is a registered trademark of Oracle Corporation, Redwood City, California.

Oracle Rdb, Oracle RMU and Rdb7 are trademarks of Oracle Corporation, Redwood City, California.

Open VMS is a trademark of Digital Equipment Corporation.

All other company or product names are used for identification purposes only and may be trademarks of their respective owners.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited. If this software is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions for commercial computer software and shall be deemed to be Restricted Rights software under Federal law, and as set forth in subparagraph (c) (1) (ii) of DFARS 252.227-7013, Rights in Technical Data and Computer Software (October 1988).

This material or any portion of it may not be copied in any form or by any means without the express prior written permission of Oracle Education Group of Oracle Corporation. Any other copying is a violation of copyright law and may result in civil and/or criminal penalties.

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with "Restricted Rights," as defined in FAR 52.227-14, Rights in Data - General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error free.

Oracle7, Oracle8, Oracle Parallel Server, Oracle Application Display Manager, Oracle Applications, Oracle Applications Window Manager, Oracle Engineering, Oracle Forms, Oracle Reports, Oracle Browser, Oracle Data Query, Oracle Document Manager, DDE Manager, Oracle Business Manager, Oracle Human Resources, Oracle Office Administrator, Oracle Office Scheduler, Oracle Office Directory, Oracle Assets, Oracle Payables, Oracle Project Accounting, Oracle Purchasing, Oracle Personnel, Oracle Open Gateway, Oracle Order Entry, Oracle Payroll, Oracle Receivables, Oracle Sales Analysis, Oracle Imaging, Oracle Inventory, Oracle*Mail, Oracle Office Mail, Oracle Manufacturing, Oracle Master Scheduling, Oracle MRP, Oracle SQL*Tutor, Oracle*Terminal, Oracle Toolkit, Oracle Work In Progress, SQL*TextRetrieval, PL/SQL, Pro*C, Report Card, Oracle Graphics, The Oracle Network Builder, Oracle Bill of Materials, Oracle BookBatch, Oracle BookBuilder, Oracle CASE, Hyper*SQL, and CASE*Generator are trademarks of Oracle Corporation.

Table of Contents

PREFACE	1
PURPOSE OF THIS HANDBOOK	1
INTENDED AUDIENCE	1
HANDBOOK CONTENTS	1
ORACLE CODASYL DBMS	2
OPERATING SYSTEM INFORMATION.....	2
RELATED MANUALS	2
ACKNOWLEDG-MENTS	2
AUTHOR.....	3
STATISTICAL INFORMATION COLLECTION	4
ACTIVE STATISTIC INFORMATION COLLECTION	4
PASSIVE STATISTIC INFORMATION COLLECTION.....	4
STATISTIC	5
COLLECTION	5
DURATION	5
PAUSING	5
STATISTIC	5
COLLECTION	5
STATISTIC INFORMATION PRESENTATION UTILITIES	6
RMU SHOW STATISTIC UTILITY.....	6
RDB PERFORMANCE MONITOR UTILITY	6
STATISTICAL INFORMATION PRESENTATION	8
STATISTIC PRESENTATION DURATION	8
PAUSING STATISTIC INFORMATION PRESENTATION.....	8
STATISTICS COLLECTION COST	9
STATISTICS PRESENTATION COST.....	9
ENABLING & DISABLING STATISTICS COLLECTION	10
DATABASE STATISTICS COLLECTION ENABLING & DISABLING	10
PROCESS-SPECIFIC STATISTICS COLLECTION ENABLING & DISABLING	10
HISTORICAL LOGICAL NAME	11
PRESERVING STATISTIC INFORMATION	12
OPERATIONAL MODES	14
ONLINE MODE	14
GLOBAL MODE	14

RECORD MODE	14
REPLAY MODE	15
PAUSED MODE	15
TYPES OF DATABASE STATISTICS	16
STEADY-STATE ANALYSIS	16
DRILL-DOWN ANALYSIS	16
DATABASE SCALING PROBLEMS	17
CHAPTER 2 - OVERVIEW	19
INVOCATION	19
COMMAND QUALIFIERS	19
PRESENTATION FACILITIES	31
BINARY INPUT FILE REPLAY	32
INFORMATION	32
SWITCHING DATABASES	36
INDIVIDUAL PROCESS MONITORING	36
CHAPTER 3 - MENUS, HELP AND PROMPTS	40
ON-SCREEN MENU	40
NAMED MENUS	43
MINI-MENUS	45
RMU SHOW STATISTIC MENU OVERVIEW	45
ONLINE HELP	48
HELP SCREEN NAVIGATION	48
PROCESS IDENTIFIER HELP	49
TOOLS AND PREFERENCES MENU	49
USER PROMPTS	49
USER RESPONSE REGION	49
USER RESPONSE TIMEOUT	50
CHAPTER 4 - PRESENTATION SCREENS	51
SCREEN DISPLAY	51
SCREEN SIZE	51
SCREEN PAGES	52
SCREEN REGIONS	53
HEADER REGION	53
COLLECTION NODE	54
INFORMATION	54
BODY REGION	54
STATUS REGION	55
SCREEN BODY DISPLAY FORMATS	56
GRAPH DISPLAY FORMAT	56
NUMERIC DISPLAY FORMAT	58
NUMERIC TABULAR DISPLAY FORMAT	60
TIME PLOT DISPLAY FORMAT	64
SCATTER PLOT DISPLAY FORMAT	66

TABULAR DISPLAY FORMAT	70
DASHBOARD DISPLAY FORMAT	71
SCREEN MIGRATION.....	73
GOTO SCREEN “BY-NAME”	73
GOTO SCREEN [->	74
RMU SHOW STATISTIC SCREEN OVERVIEW.....	75
CONFIG SCREENS.....	89
ZOOM SCREENS.....	90
ZOOM SCREEN NAVIGATION.....	91
PROCESS INFORMATION.....	92
AIJ JOURNAL INFORMATION.....	93
STORAGE AREA INFORMATION.....	94
ROW CACHE INFORMATION	95
LOCK INFORMATION	96
LOGICAL AREA INFORMATION.....	99
PAGE INFORMATION.....	101
TIME PLOT SCREEN ANALYSIS	104
SCATTER PLOT SCREEN ANALYSIS	106
SCATTER PLOT ON-SCREEN MENU DIFFERENCES.....	109
COLUMNS PER BUCKET	110
DESIGNING A CUSTOM SCREEN.....	111
CUSTOM SCREEN LOCATION.....	111
CUSTOM SCREEN METHODOLOGY	111
MANUAL CUSTOM SCREEN DESIGN	112
CUSTOM SCREEN CUSTOMIZATION	112
CUSTOM SCREEN APPEARANCE.....	113
LOG FILES.....	117
STALL MESSAGE LOG FILE	117
LOCK TIMEOUT LOG FILE.....	118
LOCK DEADLOCK LOG FILE.....	118
DBKEY ACCESS LOG FILE.....	119
PROCESS IDENTIFIER SUFFIXES.....	120
CHAPTER 5 - CONFIGURATION FILE MANAGEMENT	121
INTRODUCTION.....	121
CREATING CONFIGURATION FILES.....	121
CONFIGURATION FILE SYNTAX.....	122
IMPORTING CONFIGURATION FILES.....	123
NESTED CONFIGURATION FILES	123
CONFIGURATION VARIABLE SEMANTICS	123
VARIABLE TYPES.....	124
CONFIGURATION VALUE SEMANTICS	125
QUOTED VARIABLE EVALUATION.....	125
HIERARCHICAL VARIABLE EVALUATION.....	125

LOGICAL NAME EVALUATION.....	126
VARIABLE KEYWORD EVALUATION.....	126
PRINT & PROMPT COMMAND VARIABLES.....	126
VARIABLE VALUE REDIRECTION.....	127
LOG FILE.....	128
EXAMPLE OF A REALLY BAD CONFIGURATION FILE.....	130
CHAPTER 6 - CLUSTER STATISTIC COLLECTION.....	131
INTRODUCTION.....	131
COMMAND QUALIFIER.....	132
CONFIGURATION VARIABLE.....	132
SUPPORT.....	132
STATISTIC.....	132
COLLECTION SET.....	132
CLUSTER NODE ACCESS.....	133
SWITCHING.....	133
DATABASES.....	133
HOW CLUSTER STATISTICS COLLECTION WORKS.....	134
ENABLED VERSUS ACTIVE NODE STATUS.....	135
NETWORK.....	135
PROTOCOLS.....	135
SCREEN HEADER REGION CHANGES.....	135
BINARY FILE SUPPORT.....	136
PER-PROCESS STALL MESSAGE INFORMATION.....	136
ONLINE CLUSTER STATISTICS COLLECTION MAINTENANCE.....	137
SUMMARY.....	139
CHAPTER 7 - TOOLS.....	140
INTRODUCTION.....	140
TOOLS MENU.....	140
OPTIONS.....	140
NOTEPAD.....	148
INFORMATION.....	148
NOTEPAD EDITORS.....	149
SENDING AND/OR RECEIVING MAIL.....	149
INVOKING A.....	149
COMMAND.....	149
WAKE UP PROCESS.....	150
TERMINATE IMAGE.....	150
TERMINATE PROCESS.....	150
SET PROCESS PRIORITY.....	150
CHAPTER 8 - USER-DEFINED EVENTS.....	151
INTRODUCTION.....	151
ORACLE RDB7.....	151
EVENT DEFINITION SYNTAX.....	152
“OPERATION” CLAUSE.....	152
“STATISTIC NAME” CLAUSE.....	153

“THRESHOLD NAME” CLAUSE.....	154
“ATTRIBUTE LIST” CLAUSE.....	155
DESCRIPTION READABILITY	156
EVENT SEMANTICS.....	157
HOW USER-DEFINED EVENTS WORK	157
RUN-TIME EVENT STATUS INFORMATION	158
REAL-LIFE USER-DEFINED EVENT EXAMPLE.....	159
IDENTIFY THE OPERATION	159
IDENTIFY THE STATISTIC NAME.....	159
IDENTIFY THE THRESHOLD NAME	159
IDENTIFY THE EVENT ATTRIBUTES	159
DEFINE HOW YOU WILL BE ALERTED.....	159
WRITE THE PROGRAM.....	160
CREATE THE CONFIGURATION FILE ENTRY	160
INVOKES THE RMU SHOW STATISTIC UTILITY.....	160
SUMMARY.....	160
CHAPTER 9 - DATABASE DASHBOARD.....	161
INTRODUCTION.....	161
SECURITY.....	161
INVOCATION.....	162
DASHBOARD SCREENS DISPLAY FORMAT	162
DATABASE DASHBOARD SCREEN CONTENTS.....	163
DASHBOARD SCREENS.....	164
RUN-TIME DATABASE DASHBOARD MODIFICATION	165
RULES AND RESTRICTIONS.....	166
WARNINGS.....	167
CHAPTER 10 - ONLINE ANALYSIS	168
INTRODUCTION.....	168
DISCLAIMER.....	168
ONLINE ANALYSIS FACILITY OVERVIEW	169
FACILITY AVAILABILITY.....	169
HOW ONLINE ANALYSIS WORKS.....	169
ONLINE ANALYSIS ORIENTATION	170
ONLINE ANALYSIS THRESHOLDS.....	170
CLUSTER STATISTIC COLLECTION	170
LOGICAL AREA ANALYSIS.....	170
ANALYSIS MESSAGE TYPES	171
ONLINE ANALYSIS MESSAGES.....	171
BUFFER ANALYSIS SCREENS.....	171
TRANSACTION ANALYSIS SCREEN.....	176
AIJ ANALYSIS SCREEN	179
RUJ ANALYSIS SCREEN	190
RECOVERY ANALYSIS SCREEN.....	192
RECORD ANALYSIS SCREEN.....	195
AREA ANALYSIS SCREEN	203
LOCKING ANALYSIS SCREEN	205
INDEX ANALYSIS SCREEN.....	207

ROW CACHE ANALYSIS SCREEN	209
PROCESS ANALYSIS SCREENS.....	210
SUMMARY.....	211
CHAPTER 11 - PROBLEM DIAGNOSIS	212
INTRODUCTION.....	212
ANALYZING PROCESS INFORMATION	212
STALL MESSAGE ANALYSIS	212
INTEGRATED PAGE INFORMATION DISPLAY	214
STALL MESSAGE LOCKING ANALYSIS	216
INTEGRATED LOCK INFORMATION DISPLAY	216
LOCK DEADLOCK HISTORY SCREEN.....	220
LOCK TIMEOUT HISTORY SCREEN.....	220
PROCESS ACCOUNTING SCREEN.....	221
CHECKPOINT INFORMATION SCREEN	222
ANALYZING I/O RESOURCES	222
DATABASE PARAMETER INFORMATION SCREENS.....	222
PIO STATISTICS--DATA WRITES SCREEN	223
PIO STATISTICS--DATA FETCHES SCREEN.....	224
PIO STATISTICS--SPAM FETCHES SCREEN	226
ASYNCHRONOUS IO STATISTICS SCREEN.....	226
RECORD STATISTICS SCREEN	227
SNAPSHOT STATISTICS SCREEN.....	228
DETECTING I/O RESOURCE BOTTLENECKS.....	228
BALANCING I/O LOAD	229
CHECKING DATA DISTRIBUTION.....	231
REDUCING I/O OPERATIONS	232
IDENTIFYING EXCESSIVE I/O IN STORAGE AREAS.....	239
CHECKING VERB FAILURES.....	240
CHECKING SNAPSHOTS.....	241
ANALYZING LOCK RESOURCES.....	242
BLOCKING AST ANALYSIS	242
ANALYZING JOURNAL RESOURCES.....	244
CHECKING AIJ JOURNALS.....	244
CHECKING RUJ JOURNALS	245
AIJ STATISTICS SCREEN.....	245
AIJ JOURNAL.....	246
INFORMATION SCREEN.....	246
CHECKPOINT.....	247
STATISTICS SCREEN.....	247
ANALYZING MEMORY RESOURCES.....	249
EXCESSIVE PAGE FAULTS	249
ANALYZING.....	252
VIRTUAL MEMORY OPTIONS.....	252

Volume 1 Preface

Preface

Oracle Rdb is a general-purpose high-end database management system based on the relational data model. Applications operate in a client/server environment when accessing the database. The RMU Show Statistic utility is available for use by DBAs to monitor the operation of the application and analyze performance characteristics of the database.

PURPOSE OF THIS
HANDBOOK

This handbook describes the syntax, semantics and internal operational aspects of the RMU Show Statistic utility. The information contained in this handbook is an amalgamation of various sources of information on the utility, combined into a common frame of reference.

This handbook is not intended to be a reference manual nor should the handbook be construed as a tutorial on how to use the utility. Rather, the information contained in this handbook should be useful for anyone interested in detecting, analyzing and correcting performance problems with the database or application.

INTENDED
AUDIENCE

This handbook is intended for experienced database administrators familiar with the RMU Show Statistic utility. You should also be familiar with database management procedures and terminology.

HANDBOOK
CONTENTS

The *RMU Show Statistic DBA Handbook* is comprised of five volumes. These are the following:

- Volume 1 - Methods and Internals.
 - Volume 2 - Screen Chapters 1 through 7.
-

- Volume 3 - Screen Chapters 8 through 13.
- Volume 4 - Screen Chapters 14 through 22.
- Volume 5 - Appendix Information.

ORACLE CODASYL
DBMS

This handbook discusses Oracle Rdb. However, the RMU Show Statistic utility is almost identical for the Oracle CODASYL DBMS product DBO Show Statistic utility; in fact, it uses a common source code. Therefore, the few statistic screens that are unique to the Oracle DBMS product are included in this handbook.

OPERATING
SYSTEM
INFORMATION

The information contained in this handbook is, for the most part, operating system independent. However, on the rare occasion when an operating system specific item is discussed, the section will be appropriately highlighted.

RELATED MANUALS

For more information on the RMU Show Statistic utility, and database performance tuning information in general, please refer to the following Oracle Rdb documentation:

- Oracle Rdb RMU Reference Manual
- Oracle Rdb Guide to Database Maintenance
- Oracle Rdb Guide to Database Performance and Tuning

In addition, there are several white papers discussing RMU Show Statistic issues available in **MetaLink**. For information on the MetaLink system, please contact Oracle Client Relations or visit the Oracle Support webpage at the following location:

<http://www.oracle.com/support/>

ACKNOWLEDG -
MENTS

A special word of thanks is extended to Simon Pickering for writing the initial version of the Handbook back in 1995, upon which this new version is loosely based.

A word of thanks is also extended to Kathy Oakey, Bill Gettys, Lilian Hobbs and all the other Oracle Rdb consultants who begged and pleaded for more detailed performance analysis screens to make their job easier. Without these suggestions and ideas, the RMU Show Statistic utility would not be as powerful as it is today. I would also like to thank Norm Lastovica for his extensive knowledge of OpenVMS internals and tuning.

Finally, I have to thank Anna Logan, who wrote the *Guide to Database Performance and Tuning* manual. She diligently kept after me to keep the “online help” documentation up-to-date, even when I was adding screens to the RMU Show Statistic utility on a daily basis.

A U T H O R

If you have any comments, suggestions or corrections for this handbook, please forward them to the author at the following email address:

rjanders@us.oracle.com

Chapter 1 - Introduction

Oracle Rdb collects and maintains an immense amount of runtime statistical and analytical information about database activities as they occur over time. The RMU Show Statistic utility performs the presentation and analysis of this statistical information.

Statistical Information Collection

The database statistic information is collected and maintained in a global section that resides on each node for which the database is open. The statistic information is only physically reset when the database is closed on that node. Optionally, the statistic information can be exported for future collection.

ACTIVE STATISTIC
INFORMATION
COLLECTION

Each database application process performs the work of collecting the numerical statistical information and storing that information in the database global section. Statistical information actively collected by each process also includes the columnar numeric statistics, such as the number of transactions, as well as tabular information such as the name of the lock for which the process is currently stalled.

Processes contribute collected statistic information to the aggregated database information. This aggregation permits the RMU Show Statistic utility to display statistic information for the entire database as well as for specific application processes.

PASSIVE
STATISTIC
INFORMATION
COLLECTION

The RMU Show Statistic utility often collects statistic information for presentation only when the utility is active. This information is typically oriented towards process actions, such as checkpoint information, or after-image journal operations.

If the RMU Show Statistic utility is not active, this information is not collected.

STATISTIC
COLLECTION
DURATION

The database statistic information remains available as long as the database is open, even if the RMU Show Statistic utility is not active. The process-specific statistic information remains available only as long as the process is attached to the database.

The database statistic information is collected until the database is either implicitly or explicitly closed. A database is implicitly closed when the database open mode is automatic and the last process detaches from the database. In this mode, an active RMU Show Statistic session will keep the database open.

Issuing the RMU Close command will explicitly close a database, regardless of the database open mode.

Specifying a database close mode also controls when the database is actually closed.

Database statistics are reset to zero only when the database is closed.

Selecting the “Reset” on-screen menu option does not reset the database statistic information; rather, it resets the per-process view of the statistic information. The original database statistic information can always be retrieved.

PAUSING
STATISTIC
COLLECTION

Often, it is desirable to temporarily pause the collection of statistic information so that a snapshot view of the statistic information can be analyzed. Since Oracle Rdb is a high-performance mission-critical database management system, there is no method available to temporarily pause the collection of statistic information.

However, you can temporarily pause the presentation of the statistic information when using the RMU Show Statistic utility. The “Pause” on-screen menu option provides this capability on most types of screens. Note that there are statistic screens that cannot be paused, such as the “Process Accounting” screen.

Statistic Information Presentation Utilities

The Oracle Rdb product actually contains two statistic presentation utilities: the **RMU Show Statistic** utility and the **Rdb Performance Monitor** utility.

Starting with Oracle Rdb 8.0, the **Rdb Collection Cartridge** can also be used to retrieve statistic information.

RMU SHOW
STATISTIC
UTILITY

The RMU Show Statistic utility provides a portable command-line presentation of the statistical information. This utility works on any workstation or dumb display terminal.

The RMU Show Statistic utility is an integral component of the base Oracle Rdb product; it is not a separately bundled utility. There is no additional license requirements or fees for this utility.

The RMU Show Statistic utility is able to display only one screen at a time. To view multiple screens of information at the same time requires multiple invocations of the utility.

The RMU Show Statistic utility is often referred to as the character-cell utility.

RDB
PERFORMANCE
MONITOR UTILITY

The Rdb Performance Monitor utility, also known as the RMU Windows Statistic utility, provides a windowing interface to the RMU Show Statistic utility.

To use this interface on a PC, you must be using Microsoft Windows95 or Windows/NT, and have the RMUwin utility installed. The RMUwin and Rdb Performance Monitor utilities are part of the Oracle DBApack, which is a separately licensed Oracle product and not included in the base Oracle Rdb product.

The Rdb Performance Monitor utility does not run on OpenVMS; you must use the RMU Show Statistic utility instead.

The largest benefit of the Rdb Performance Monitor utility is that, by using an appropriate display terminal or workstation monitor, you can have several windows, or screen displays, active simultaneously.

Note that you can also launch the Rdb Performance Monitor utility through the RMUwin utility directly.

The Rdb Performance Monitor utility display formats are similar to the character-cell display formats. However, not all displays are available using the windows interface.

Since the Rdb Performance Monitor utility functionality is a subset of the RMU Show Statistic utility, the majority of the discussions in this handbook apply to the RMU Show Statistic utility.

Statistical Information Presentation

The RMU Show Statistic utility performs the actual presentation of the statistical information. As of Oracle Rdb 7.0, the runtime processes do not perform any of the presentation formatting; the runtime processes no longer perform any screen formatting. This significantly improves the runtime performance and transaction throughput of the application.

There can be an unlimited number of RMU Show Statistic utilities running on a single database. Each process using the RMU Show Statistic utility displays the same statistic information, but can display different screens independently.

Normally, the RMU Show Statistic utility displays statistic information for the current node only. However, statistic information can also be displayed for all nodes on which the database is open, or any set of open nodes.

STATISTIC
PRESENTATION
DURATION

Presentation of the database statistic information occurs until one of the following events occurs: the time specified by the UNTIL qualifier is reached, until you select the “Exit” on-screen menu option, or until the database is explicitly closed using the RMU Close utility.

The “Reset” on-screen menu option stops the current collection of statistics and starts a new collection of statistic information. The statistics are reset for that process only; each active RMU Show Statistic utility can reset the statistics independently of each other.

The “Unreset” on-screen menu option is available to revert back to the values collected since the database was opened. If the database open mode is manual, you can also exit and restart the utility to display the original statistic information.



Oracle Corporation recommends you use the “Unreset” on-screen menu option. There is no way to unreset the values back to a previously reset value.

If you want to keep the database open for an extended period of time to observe processing activity, you can explicitly open the database using the RMU Open utility.

PAUSING
STATISTIC
INFORMATION
PRESENTATION

The “Pause” on-screen menu option temporarily pauses the display of the statistic information; it does not actually affect the collection of the statistic information, however.

Selecting the “Pause” on-screen menu option will resume the display of the statistic information.

Note that when replaying a binary input file, you can also pause the replaying of the file. This is a separate operation from pausing the display. If you pause the display when using a binary input file, the file will continue to be processed but the statistic information will not be displayed.

STATISTICS
COLLECTION COST

The cost of collecting the statistic information varies depending on the size of the database and the operations being performed by the application. The statistic collection region in the database global section is based on a large number of factors, including number of users, storage areas, database files such as after-image journals and recovery-unit journals, logical areas and row caches. In general, however, collecting statistics information incurs a 7-10% overhead.

Prior versions of Oracle Rdb incurred a higher runtime statistic collection cost. Oracle Rdb 7.0 has optimized the runtime statistic collection code to minimize the runtime impact on the application.

Obviously, the actual impact on the system and database varies greatly depending on the specific database and application. However, without the ability to use the RMU Show Statistic utility, you will never know how well your application is performing. In most cases, the benefits provided by the information greatly outweigh the cost of collecting the information.

STATISTICS
PRESENTATION
COST

The cost of using the RMU Show Statistic utility varies depending on the screen being displayed and the screen refresh rate being used. Typically, when using Oracle Rdb 7.0, a three-second screen refresh rate incurs approximately 2-3% overhead, while a one-second screen refresh rate incurs approximately 5-10% system overhead. Using a one-tenth-second screen refresh rate typically incurs approximately 25-40% overhead.

Oracle Rdb 7.0 has been optimized for both memory utilization and code efficiency. Even though the RMU Show Statistic utility provides additional functionality over previous versions, it actually consumes less CPU and has a smaller code path than prior versions.



Note that presenting logical area statistics consumes a large amount of virtual memory. This large use of virtual memory may cause the utility to perform excessive page faulting. Be sure to review process quotas when initially using the utility. Use of the NOLOGICAL_AREA qualifier affects whether or not the utility displays logical area statistic information.

Enabling & Disabling Statistics Collection

You can enable and disable statistics collection for either the entire database, or for specific processes.

DATABASE
STATISTICS
COLLECTION
ENABLING &
DISABLING

To enable statistics collection for a database, use the `STATISTICS COLLECTION IS ENABLED` clause of the SQL `CREATE DATABASE`, `ALTER DATABASE`, or `IMPORT` statements.


The following `ALTER DATABASE` statement enables statistics collection for the `MF_PERSONNEL` database:

```
SQL> ALTER DATABASE FILENAME MF_PERSONNEL  
cont> STATISTICS COLLECTION IS ENABLED;
```

You can disable statistics collection for a database by using the `STATISTICS COLLECTION IS DISABLED` clause of the SQL `CREATE DATABASE`, `ALTER DATABASE`, or `IMPORT` statements.

The following `ALTER DATABASE` statement disables statistics collection for the `MF_PERSONNEL` database:

```
SQL> ALTER DATABASE FILENAME MF_PERSONNEL  
cont> STATISTICS COLLECTION IS DISABLED;
```

 By default, statistics collection is enabled for a database. Oracle Corporation recommends that you leave the statistics collection enabled, so that the information is available should you need it.

When statistics collection is disabled for a database, no statistics are displayed for any of the processes attached to the database.

To minimize virtual memory consumption, it is possible to disable local process statistics collection.

PROCESS -
SPECIFIC
STATISTICS
COLLECTION
ENABLING &
DISABLING

If statistics collection is disabled for the database, you cannot enable statistics collection for specific processes.

If statistics collection is enabled for the database, it is possible to disable the collection of database statistics for a specific process. By default, the collection of database statistics is enabled for each process on a node. When you disable database statistics for a process, Oracle Rdb does not collect any statistics for that particular process; the statistics displayed on the various screens will not include information on processes for

which statistics have been disabled. Disabling statistics is useful for static, performance-critical applications that have been previously tuned and do not need the information provided by the statistics collection package.

Note that, even though a process may not be actively collecting statistic information, the process will still appear in the process-related screens, such as the “Checkpoint Information” screen. These types of screens use passive information collection mechanisms, performed by the RMU Show Statistic utility, not the process.

You can disable the collection of statistics for all processes in one of two ways:

- By setting the `RDM$BIND_STATS_ENABLED` logical name to the value “0” for each process; defining `RDM$BIND_STATS_ENABLED` as a group or system logical can help simplify this task.
- By disabling statistics collection for the database itself.

You disable the collection of database statistics for a process by setting the `RDM$BIND_STATS_ENABLED` logical name to the value of “0”, as follows:

```
$ DEFINE RDM$BIND_STATS_ENABLED 0
```

To enable the collection of database statistics for a process in which the collection of database statistics is disabled, define the logical name specifying the value “1”, or deassign the logical name as follows:

```
$ DEFINE RDM$BIND_STATS_ENABLED 1
```

Or, deassign the logical name:

```
$ DEASSIGN RDM$BIND_STATS_ENABLED
```

HISTORICAL
 LOGICAL NAME

Prior to Oracle Rdb 7.0, the `RDM$BIND_STATS_DISABLED` logical name was used to enable and disable statistics collection on a per-process basis. The following information applies to prior versions of Oracle Rdb, but can also be used in Oracle Rdb 7.0.1.0 for backwards compatibility.

You can disable the collection of statistics for all processes in one of two ways:

- By setting the `RDM$BIND_STATS_DISABLED` logical name to the value “1” for each process; defining `RDM$BIND_STATS_DISABLED` as a group or system logical can help simplify this task
- By disabling statistics collection for the database itself.

You disable the collection of database statistics for a process by setting the `RDM$BIND_STATS_DISABLED` logical name to the value of “1”, as follows:

```
$ DEFINE RDM$BIND_STATS_DISABLED 1
```

To enable the collection of database statistics for a process in which the collection of database statistics is disabled, define the logical name specifying the value “0”, or deassign the logical name as follows:

```
$ DEFINE RDM$BIND_STATS_DISABLED 0
```

Or, deassign the logical name:

```
$ DEASSIGN RDM$BIND_STATS_DISABLED
```

Preserving Statistic Information¹

Ordinarily, the accumulated database statistic information is discarded when the database is closed. This makes it very difficult to accumulate statistic information for an extended period of time, such as a month or more, where the database is closed several times during that period.

The RMU Close command includes the STATISTICS qualifier. The STATISTICS qualifier indicates that the current statistic information should be preserved. The default is NOSTATISTICS, which indicates statistic information is NOT preserved on database close, as is normally the case.

The RMU Open command also includes the STATISTICS qualifier. The STATISTICS qualifier indicates that previously saved statistic information should be loaded when the database is opened. The default is NOSTATISTICS, which indicates statistic information is NOT loaded on database open, as is normally the case. The STATISTICS qualifier also indicates that the database monitor should periodically checkpoint the statistic information to disk.

The statistic information is stored in a node-specific database file, located in the database directory. The file is named “database_node.RDS”. For example, the MF_PERSONNEL database open on node MYNODE would use a statistic file named MF_PERSONNEL_MYNODE.RDS.

Note that clusterwide statistic information is NOT stored in the statistic file. This allows you to decide on which nodes the statistic information should be initially loaded at database open time.

The statistic information files may not be renamed or copied as they contain node specific information. The statistic information files may be deleted, if they are no longer needed; they are not required in order to open a database.

¹ Available in Oracle Rdb 7.0.2.

The statistic files will not be loaded if the physical schema of the database has changed since the statistic file was created. This means that the addition or deletion of storage areas, logical areas (tables and/or indexes) and record caches will invalidate the statistic information files. This restriction prevents incorrect statistic information from being loaded when intervening physical changes occurred to the database. Closing the database will update the statistic information files and make them valid again.

The STATISTICS qualifier cannot be specified in conjunction with the RMU Close Cluster command. To preserve the statistics information for a database open on a cluster, you must specifically close the individual nodes.

If you specified the STATISTICS qualifier on the RMU Open command, then the statistics information will be automatically preserved in the event of abnormal database closure (such as DBR failure).



The Oracle 8.0 RMU Open command includes several keywords for the STATISTICS qualifier. However, the STATISTICS qualifier still works as described above.

Operational Modes

The RMU Show Statistic utility contains the following operational modes: online, record, replay and paused.

ONLINE MODE

In online mode, you display the runtime statistic information for the database. This is the default operational mode, and the mode which is most commonly used. In online mode, database activity is monitored as it occurs in realtime. The utility displays new statistics every 3 seconds, unless you specify a preferred time interval with the TIME qualifier. You can change the recording interval at any time during an interactive session; select the “Set_rate” on-screen menu option and specify the desired interval.

You can display the statistics on your terminal by using the INTERACTIVE qualifier, or you can record the statistics in a binary file by using the OUTPUT qualifier. Oracle Rdb normally collects statistics for a database as long as the database is open, but you can use the UNTIL qualifier to specify a time in the future when statistics collection will stop. The UNTIL qualifier is typically used with the OUTPUT qualifier in a batch job.

GLOBAL MODE²

Essentially the same as the online mode, this mode indicates that process global statistic information is available for all processes. This mode occurs when the database is opened using the STATISTICS=PROCESS_GLOBAL qualifier on the RMU Open command. For more information on this mode, please refer to the “Individual Process Monitoring” section in Chapter 2.

RECORD MODE

You can record statistics in a binary file by using the OUTPUT qualifier. The binary file does not produce a legible printed listing. You can read the binary file using replay mode; use the RMU Show Statistic utility and specify the name of the binary file as an argument to the INPUT qualifier. You can also write your own program to reformat the file.

Of course, you can also display the statistic information while you record the same information, if you so desire.

Oracle Rdb normally records statistics for a database as long as the database is open, but you can use the UNTIL qualifier to specify a time in the future when statistics collection will stop. The UNTIL qualifier is typically used with the OUTPUT qualifier in a batch job.

² Available in Oracle Rdb 8.0.

You can use the `NOINTERACTIVE` qualifier to suppress the interactive display. This is useful if you want to generate a binary statistics output file for later reference, but do not want an online display while statistics are being collected.

Note that batch jobs require the `NOINTERACTIVE` qualifier.

More information on the binary file can be found in Appendix A.

REPLAY MODE

In replay mode, database activity previously recorded in a binary file is read and displayed on your terminal. The statistics would have been recorded in online mode using the `OUTPUT` qualifier.

To replay previously recorded statistics, use the `INPUT` qualifier and specify the name of the file that contains the statistics. You can then use the arrow keys and the menus to change the display format or the screen, just as in online mode. By using replay mode, you can view a sampling of statistics many times and examine various aspects of the sampling. This is valuable if you find something on one screen, and you want to look at another screen to confirm your analysis.

You can change the rate of the display by specifying the desired rate with the `TIME` qualifier or use the “Set_rate” on-screen menu choice. Changing the display rate in replay mode is not the same as in online mode. For example, suppose you gathered statistics into a file in online mode using the default interval of 3 seconds. When you replay statistics from the file, changing the display rate changes the rate at which those 3 second intervals are displayed; it does not change the statistics themselves.

PAUSED MODE

In paused mode, the display of statistic information is temporarily suspended, even though statistic collection still occurs. When replaying statistics from a file, the actual reading of the file is also paused.

Pause mode is obtained by selecting the “Pause” on-screen menu option. Selecting “Pause” again resumes statistic presentation and reverts back to the previous operational mode.

Note that when replaying a binary input file, you can also pause the replaying of the file. This is a separate operation from pausing the display. If you pause the display when using a binary input file, the file will continue to be processed but the statistic information will not be displayed.

Types of Database Statistics

There are two basic types of statistics that are displayed by the RMU Show Statistic utility:



Statistics that provide general information about what is happening with your database, such as transactional access patterns and the system workload.



Statistics that are tied to parameters and database attributes whose settings you can modify to improve database performance, such as the PIO statistic where the buffer pool overflows and a modified page must be written back to the database.

The way you interpret these statistics depends on how your database is structured and how it is actually used. If you are confident that your database is well-designed, you should use the steady-state approach to analyzing the performance characteristics of your database. Otherwise, the drill-down approach to analyzing the performance characteristics of your database should be used.

STEADY-STATE
ANALYSIS

In the steady-state approach to performance analysis, you monitor the interesting statistics on a regular basis until you become familiar with the normal performance and operation of the database. You then monitor the database on a continued periodic basis trying to identify deviations from the baseline expectations, and adjust either the database design, the application or system parameters to resolve the problem and eliminate the performance deviations.

DRILL-DOWN
ANALYSIS

In the drill-down approach to performance analysis, you monitor the various statistic screens trying to identify anything that might indicate a performance problem. Often, it is possible to identify a problem in one statistic and resolve it fairly easily. The goal of the drill-down analysis is to achieve a benchmark or baseline so that you can use the steady-state analysis approach.

The RMU Show Statistic utility “Online Analysis” facility is available to help you perform drill-down analysis. See Volume 1, Chapter 10 for more information on this facility.

It may be necessary to drill down to a specific process to separate individual statistic characteristics from other processes. This process drill-down is accomplished using the “Process Monitoring” facility.

DATABASE
SCALING
PROBLEMS

Quite often, an application is written when the database is relatively young and small. While the database is small, performance of the application might be acceptable. It is only when the database grows in size that performance of the application degrades to unacceptable levels.

This is known as scaling the database and application. In this case, you need to gather statistics on the database as it grows and compare the differences in transaction rates over time. Comparing the statistics should help you highlight the cause of the performance problems. Also, by comparing the statistics from different database sizes, you can identify the database size where performance degradation occurs.

Getting Started

Chapter 2 - Overview

This chapter provides an overview of the RMU Show Statistic utility and how to get started analyzing the performance characteristics of the database and application.

INVOCATION

The general format for invoking the CLI-based version of the Show Statistic utility is the following:

```
$ RMU Show Statistic [ optional qualifiers ] [ optional database_name ]
```

Note that the *database_name* parameter is optional only if you specify the INPUT qualifier. Otherwise, the *database_name* parameter is always required.

You launch the GUI-based version of the Rdb Performance Monitor from the RMUwin utility of the Oracle DBApack.

Note that, unless explicitly stated, discussions in this handbook are primarily directed towards the CLI-based RMU Show Statistic utility. The CLI-based utility is more portable and provides a larger variety of screens and analysis capability.

COMMAND
 QUALIFIERS

The RMU Show Statistic utility contains a large number of optional command qualifiers. The majority of the command qualifiers can be superseded by configuration variables, which are stored in a persistent disk file. Also, most of the command qualifier values can be modified at runtime.

The following optional command qualifiers are available:

/ALARM The `ALARM=interval` qualifier limits the display of stall messages to those stalls that exceed the specified number of seconds. An alarm interval of "0" seconds indicates

that all stall messages should be displayed, since by definition a stall is any operation that does not immediately succeed.

You can also specify the alarm interval at runtime using the “Alarm” on-screen menu option.

For additional information on the use of the ALARM qualifier, see the expanded description of the [NO]NOTIFY qualifier.

The default is ALARM=0 seconds.

The corresponding configuration variable name is ALARM.

/[NO]BROADCAST This qualifier specifies whether or not operating system broadcast messages should be displayed on the screen status region as they are received. This is typically useful for knowing when new mail has arrived.

This qualifier cannot be modified at runtime.

The default is BROADCAST.

The corresponding configuration variable name is BROADCAST.

/[NO]CLUSTER The [NO]CLUSTER=*[node_list]* qualifier specifies the optional comma-separated list of cluster nodes from which statistic collection is to be performed. If the comma-separated list of cluster nodes is not specified, this indicates that statistic collection is to be performed on all nodes where the database is currently open.

The CLUSTER qualifier cannot be used if the INPUT qualifier is specified.

For additional information on the CLUSTER qualifier, please refer to Volume 1, Chapter 6.

The default is NOCLUSTER, which indicates to perform statistic collection from the local node only.

The corresponding configuration variable name is CLUSTER_NODES.

/CONFIGURE The CONFIGURE=*configuration_file name* qualifier specifies the name of the configuration file.

For additional information on the CONFIGURE qualifier, please refer to Volume 1, Chapter 5.

`/[NO]CYCLE` The `[NO]CYCLE=interval` qualifier specifies the number of seconds to display each screen in the current menu, thereby causing the screens to cycle from one to another. A value of “0” indicates that screen cycling is disabled, which is comparable to `NOCYCLE`.

This qualifier can be modified at runtime using the “Stop screen cycling” or “Start screen cycling” Tools menu options.

The default is `NOCYCLE`, which indicates that screen cycling is disabled.

The corresponding configuration variable name is `CYCLE`.

`/[NO]DBKEY_LOG` The `[NO]DBKEY_LOG=log_file name` qualifier specifies the name of the output file to which `DBKEY` access information is to be logged.

This qualifier can be modified at runtime using the “Stop `DBKEY` logging” or “Start `DBKEY` logging” Tools menu options.

The default is `NODBKEY_LOG`, which indicates that `DBKEY` access logging is disabled.

The corresponding configuration variable name is `DBKEY_LOG`.

For more information on the `DBKEY_LOG` qualifier, please refer to the “`DBKEY` Access Log File” Section.

`/[NO]DEADLOCK_LOG` The `[NO]DEADLOCK_LOG=log_file name` qualifier specifies the name of the output file to which lock deadlock history information is to be logged.

This qualifier can be modified at runtime using the “Stop lock deadlock logging” or “Start lock deadlock logging” Tools menu options.

The default is `NODEADLOCK_LOG`, which indicates that lock deadlock logging is disabled.

The corresponding configuration variable name is `DEADLOCK_LOG`.

Note: This qualifier does not have a “`LOCK_`” prefix. Do not confuse the `DEADLOCK_LOG` qualifier with the `LOCK_TIMEOUT_LOG` qualifier (described below). Due to the manner in which OpenVMS parses the `RMU Show Statistic` utility command qualifiers, specifying the `LOCK_DEADLOCK_LOG` qualifier will incorrectly result in the `LOCK_TIMEOUT_LOG` qualifier being used instead.



For more information on the DEADLOCK_LOG qualifier, please refer to the “Lock Deadlock Log File” Section.

/[NO]HISTOGRAM This qualifier specifies whether to display the initial statistics screen in the numbers display mode or the graph display mode. The HISTOGRAM qualifier specifies the graph display mode. The NOHISTOGRAM qualifier specifies the numbers display mode, which is normally the best mode to use since it provides the most detailed information.

This qualifier can be modified at runtime using the “Graph” and “Numbers” on-screen menu options.

The default is NOHISTOGRAM, which indicates that the numeric columnar display format is to be used.

The corresponding configuration variable name is HISTOGRAM.

/INPUT The INPUT=*file name* qualifier specifies the pre-recorded binary file from which you can read the statistics. This file must have been created by an earlier RMU Show Statistic session that specified the OUTPUT qualifier.

Note that you do not specify a database name when you use the INPUT qualifier; for example:

```
$ RMU /SHOW STATISTIC /INPUT=STATS.DAT
```

This qualifier cannot be modified at runtime.

There is no default qualifier value.

There is no corresponding configuration variable name.

Please refer to the “Binary Input File Replay” section of this chapter for more detailed information on the “Binary File Replay” facility.

/[NO]INTERACTIVE This qualifier specifies whether or not to display the statistic information to the terminal. Typically, the INTERACTIVE qualifier is available for use when submitting the RMU Show Statistic utility in a batch job, where statistic information output is not desired.

The NOINTERACTIVE qualifier cannot be used if the INPUT qualifier is specified.

This qualifier cannot be modified at runtime.

The default is INTERACTIVE.



The corresponding configuration variable name is INTERACTIVE.

`NO]LOCK_TIMEOUT_LOG` The `[NO]LOCK_TIMEOUT_LOG=log_file_name` qualifier specifies the name of the output file to which lock timeout history information is to be logged.

This qualifier can be modified at runtime using the “Stop lock timeout logging” or “Start lock timeout logging” Tools menu options.

The default is `NOLOCK_TIMEOUT_LOG`, which indicates that lock timeout logging is disabled.

The corresponding configuration variable name is `LOCK_TIMEOUT_LOG`.

For more information on the `LOCK_TIMEOUT_LOG` qualifier, please refer to the “Lock Timeout Log File” section.

`/[NO]LOG` Logs interesting RMU Show Statistic utility messages, such as the creation of a binary statistics file, to your `SYSSOUTPUT` file. This binary statistics file is created only if you have used the `OUTPUT` qualifier. If you use the `NOLOG` qualifier, no operations will be logged to your `SYSSOUTPUT` file.


If you use the `INTERACTIVE` qualifier, the `LOG` qualifier is ignored.

The default is the current setting of the DCL verify switch. See `HELP SET VERIFY` in `DCL HELP` for more information on changing the DCL verify switch.

This qualifier cannot be modified at runtime.

There is no corresponding configuration variable name.

`/[NO]LOGICAL_AREA3` This qualifier specifies that logical area statistics are to be presented. The presentation of logical area statistics requires a significant amount of virtual memory.

 The collection of logical area statistic information requires, on average, approximately 13k bytes of virtual memory. If you do not wish to view the logical area statistic screens, then using the `NOLOGICAL_AREA` qualifier is recommended. Also, the amount of virtual memory required is based on the largest logical area identifier, not the actual number of logical areas. Therefore, a database containing 50 logical areas whose largest logical area identifier is 1234 would require approximately 16Mb of virtual memory to present the logical area information.

³ Available in Oracle Rdb 7.0.2.0.

Note that using the NOLOGICAL_AREA qualifier may affect the “PageInfo” on-screen menu option availability.

Be careful when abbreviating the NOLOGICAL_AREA qualifier. Do not inadvertently specify the NOLOG qualifier instead of the NOLOGI qualifier.

The default is LOGICAL_AREA, which indicates that logical area statistics are to be presented.

This qualifier cannot be modified at runtime.

There is no corresponding configuration variable.

/[NO]NOTIFY This qualifier specifies the set of operator classes to be used when a process has exceeded the stall threshold specified by the ALARM qualifier.

The full syntax for the qualifier is:

`/[NO]NOTIFY [= ([no] all | operator-classes)]`

The list of valid operator classes can contain any or all of the following keywords: CENTRAL CLUSTER DISKS SECURITY OPER1 OPER2 OPER3 OPER4 OPER5 OPER6 OPER7 OPER8 OPER9 OPER10 OPER11 OPER12.

The default for the NOTIFY qualifier is NONNOTIFY. The NOTIFY qualifier is only used in conjunction with the ‘Alarm’ screen menu option while the Stall Messages screen is displayed.

The following rules apply to the qualifier:

<u>Qualifier</u>	<u>Description</u>
/NONNOTIFY	Disables operator notification.
/NOTIFY	Enables CENTRAL and CLUSTER operator classes.
/NOTIFY=All	Enables all operator classes.
/NOTIFY=NoAll	Disables operator notification.
/NOTIFY=operator-classes	Enables all specified operators; e.g. /NOTIFY=(OPER1, OPER2).

The specified system operator(s) will be notified whenever the alarm threshold is initially exceeded. For example, if three processes exceed the alarm threshold, the specified operator(s) will be notified only once. If another process subsequently exceeds

the alarm threshold while the other processes are being displayed, the specified system operator(s) will not be notified. However, if the longest-duration stall is resolved and a new process then becomes the newest stall to exceed the alarm threshold, then a new notification is sent.

The operator notification message will appear similar to the example:

```
%%%%%%%%%% OPCOM dd-mmm-yyyy hh:mm:ss.cc %%%%%%%%%% (from node
MYNODE at dd-mmm-yyyy hh:mm:ss.cc)
Message from user USERNAME on MYNODE
Oracle Rdb Database DISKS:[directory]MF_PERSONNEL.RDB:1 Event Notification
Process 202000A1:1 exceeded 5 second stall: waiting for record 1:2:3 (EX)
```

In order to receive operator notification messages the following three OpenVMS DCL commands must be issued:

```
$SET TERM /BROADCAST
$SET BROADCAST=OPCOM
$REPLY /ENABLE=(operator-classes)
```

Note that the list of operator classes specified by the *operator-classes* parameter must match those classes specified by the NOTIFY qualifier.

To simplify establishing an alarm interval for the “Stall Messages” screen, the ALARM interval qualifier has been added. The ALARM qualifier allows the setting of the desired alarm interval, in seconds, from the command line; this is useful when submitting the RMU Show Statistic utility as a batch job. The default ALARM qualifier is “0” which effectively disables the feature.

This qualifier cannot be modified at runtime.

The default is NONOTIFY, which indicates that operator notification is disabled.

The corresponding configuration variable name is NOTIFY.

/OPTIONS The *OPTIONS=keyword_list* qualifier specifies various processing options, typically used in conjunction with the OUTPUT qualifiers.

The following keywords are currently supported:

<u>Keyword</u>	<u>Description</u>
BASE	This keyword indicates that only the base set of statistics should be collected; this is the default Options keyword. The base set of statistics is the statistics that were collected prior to Oracle Rdb Version 6.1 (no by-area statistics are collected when you select the BASE keyword). You cannot

<u>Keyword</u>	<u>Description</u>
	specify NOBASE.
[NO]AREA	<p>This keyword indicates whether or not the by-area statistics information is to be collected in addition to the base set of statistics information.</p> <p>When the Area keyword is specified, statistics for all existing storage areas are written to the binary output file; you cannot select specific storage areas whose statistic information is to be collected.</p>
[NO]ALL	<p>This keyword indicates whether or not all collectible statistics (all statistics for all areas) should be collected. The All keyword indicates that all statistics information is to be collected, while the NOALL keyword indicates that only the base statistics information is to be collected.</p>
[NO]CONFIRM	<p>This keyword indicates whether or not you wish to confirm a request to exit the RMU Show Statistic utility. This option is primarily used to prevent accidental exit from the utility. The default is NOCONFIRM.</p>
[NO]ROW_CACHE	<p>This keyword indicates whether or not you wish to collect statistic information for all row caches in the databases. Collection of row cache statistic information may require access to all active row caches in the database and may impact system performance. The default is NOROW_CACHE.</p>
[NO]UPDATE	<p>This keyword indicates whether or not you wish to be allowed to update the database dashboard information. The default is NOUPDATE.</p>
[NO]VERBOSE	<p>This keyword indicates whether or not stall messages should be written verbosely to the stall log file. For more information on the effect of this keyword, please refer to the STALL_LOG qualifier description. The default is NOVERBOSE.</p>

The default keywords for the OPTIONS qualifier are:

/OPTIONS=(BASE, NOCONFIRM, NOROW_CACHE, NOUPDATE, NOVERBOSE)

If you specify a list of keywords, a keyword can negate the meaning of a keyword that appears earlier in the list. For example:

<u>Qualifier</u>	<u>Description</u>
/OPTIONS=(All, Noall)	qualifier selects Base statistics.
/OPTIONS=(Area, Base)	qualifier selects Base statistics.
/OPTIONS=(All, Noarea)	qualifier selects Base statistics.
/OPTIONS=(Noarea, Area)	qualifier selects Base and Area statistics.
/OPTIONS=(Noarea, All)	qualifier selects all statistics.

The following command creates the binary output file with base statistics for all the storage areas in the MF_PERSONNEL database:

```
$ RMU Show Statistic /OUTPUT=main.stats /OPTIONS=(Base, Area)
MF_PERSONNEL
```

You can replay the statistics as follows:

```
$ RMU Show Statistic /INPUT=main.stats
```

The size of the by-area statistics output largely depends on the total number of storage areas in the database, including reserved storage areas. If the database contains a large number of storage areas, it may not be advisable to use the OPTIONS=Area qualifier.

The binary output file format generated by the RMU Show Statistic utility has changed significantly in Oracle Rdb V7.0.

This qualifier cannot be modified at runtime.

There is no corresponding configuration variable name.

/[NO]OUTPUT The [NO]OUTPUT=*file name* qualifier is used to direct statistical binary information to a file. The output is a formatted binary file and does not produce a legible printed listing. To read the output, you must use the RMU Show Statistic utility with the INPUT qualifier.

The OUTPUT qualifier cannot be used if the INPUT qualifier is specified.

The default is NOOUTPUT, which indicates that a binary output file is not created.

The corresponding configuration variable name is OUTPUT.

`/[NO]PROMPT_TIMEOUT` The `[NO]PROMPT_TIMEOUT=interval` qualifier specifies the maximum number of seconds that user prompts are to wait for a reply. User prompts include named menus; on-screen menus never have a user response timeout since they are always active.

The default is `PROMPT_TIMEOUT=60` seconds.

The corresponding configuration variable name is `PROMPT_TIMEOUT`.

`/REOPEN_INTERVAL` The `REOPEN_INTERVAL=interval` qualifier specifies the number of seconds after which the file specified by the `OUTPUT` qualifier is closed and a new binary output file opened. This prevents each binary output file from becoming too large.

There is no default qualifier.

The corresponding configuration variable name is `REOPEN_INTERVAL`.

`/RESET` The RMU Show Statistic utility displays the total amount of database activity since the database was opened. If you used RMU Show Statistic interactively prior to Oracle Rdb V7.0, you could select the Reset option from the first screen's on-screen menu to reset the values and display the activity since the values were reset. If the RMU Show Statistic utility was used in batch mode, however (by specifying the `NOINTERACTIVE` command qualifier), you could not reset the values. You could display only the total amount of database activity since the database was opened.

When you specify the Reset qualifier, the statistics values are reset before the initial display (if the utility is available for use interactively) or before any statistics are written to the output file (if the utility is available for use in batch).

The corresponding configuration variable name is `RESET`.

`/SCREEN` The `SCREEN=screen title` qualifier enables you to specify an initial screen for the RMU Show Statistic utility. For example:

```
$RMU Show Statistic /SCREEN="Stall Messages" database_name
```

causes the Stall Messages screen to be the first screen displayed. You can use any unique portion of the desired screen name; the previous example could have used `/SCREEN="stall"`.

The following restrictions apply to the `SCREEN` qualifier:

- The screen name is case insensitive.
- The screen name (portion) must match exactly.
- If the specified screen cannot be found, then the “Summary IO Statistics” screen is displayed without any error message.
- Use quotes around the screen name if the screen name has embedded spaces.
- If you specify the NOINTERACTIVE qualifier, then the SCREEN qualifier is ignored.
- You cannot select the by-lock or by-area screens using the SCREEN qualifier.

By default, the first screen is the “Summary IO Statistics” screen.

The corresponding configuration variable name is SCREEN.

`/[NO]STALL_LOG` The `[NO]STALL_LOG=log_file name` qualifier specifies the name of the output file to which process stall messages should be logged.

By default, stall messages are only written once as they occur. However, you can also have stall messages logged for their entire duration using the `OPTION=VERBOSE` qualifier or the `STALL_LOG_VERBOSE` configuration variable.

This qualifier can be modified at runtime using the “Stop stall logging” or “Start stall logging” Tools menu options.

The default is `NOSTALL_LOG`, which indicates the stall messages are not written to the log file.

The corresponding configuration variable name is `STALL_LOG`.



For more information on the `STALL_LOG` qualifier, please refer to the “Stall Message Log File” section. Also, Appendix J contains interesting information regarding how to send stall messages to an OpenVMS mailbox.

`/TIME` The `TIME=interval` qualifier specifies the statistics collection interval in seconds. This qualifier reflects how frequently the statistic information on the screen is refreshed.

If you omit the `TIME` qualifier, a sample collection will be made every 3 seconds. The integer has a normal range of 1 to 180 (1 second to 3 minutes). However, if you specify a negative number for the `TIME` qualifier, the RMU Show Statistic utility interprets

the number as hundredths of a second. For example, TIME=-20 specifies an interval of 20/100 or 1/5th of a second.

If you are running the RMU Show Statistic utility interactively, it updates the screen display at the specified interval.

This qualifier can be modified at runtime using the “Set_rate” on-screen menu option.

If you are also using the OUTPUT qualifier, a binary statistics record is written to the output file at the specified interval. A statistics record is not written to this file if no database activity has occurred since the last record was written.

If you are also using the “Stall Message Logging” facility specified by the STALL_LOG qualifier, a stall message is written to the output file at the specified interval.

If you are also using the “Lock Deadlock Logging” facility specified by the DEADLOCK_LOG qualifier, a lock deadlock message is written to the output file at the specified interval.

If you are also using the “Lock Timeout Logging” facility specified by the LOCK_TIMEOUT_LOG qualifier, a lock timeout message is written to the output file at the specified interval.

If you are also using the “DBKEY Access Logging” facility specified by the DBKEY_LOG qualifier, a DBKEY access message is written to the output file at the specified interval.

The corresponding configuration variable name is REFRESH_INTERVAL.

/UNTIL The UNTIL=*date-time* qualifier specifies the absolute time in the future when the statistics collection should end. When this point is reached, the RMU Show Statistic utility terminates and control returns to the DCL command level. When the RMU Show Statistic utility is executed in a batch job, the batch job will terminate at the time specified.

The UNTIL qualifier accepts OpenVMS Version 5.0 and higher date and time strings, as well as international dates. See the OpenVMS *RTL Library (LIBS) Manual* for more information. The date and time strings specified must be enclosed within quotation marks because they can contain spaces and other DCL syntax characters such as commas. A colon separator between the date and time field is no longer accepted.

An example of using the UNTIL qualifier follows:

```
$ DEFINE LIB$DT_INPUT_FORMAT -
  " !MAU !DB, !Y4 !H04: !M0: !S0. !C2"
$ RMU /Show Statistic /UNTIL="JUNE 16, 1998 17:00:00.00" -
  MF_PERSONNEL
```

This example stops execution of the RMU Show Statistic utility at 5 P.M. on June 16, 1998. You can omit the date if you wish to use the default of today's date. Be careful to specify a date that is in the future; a date specified in the past will cause the RMU Show Statistic utility to immediately terminate.

If you do not use the UNTIL qualifier, the RMU Show Statistic utility continues until you terminate it manually. In an interactive session, you can terminate the utility by pressing Ctrl/Z or by selecting "Exit" from the on-screen menu. When you are running the RMU Show Statistic utility with the NOINTERACTIVE qualifier from a terminal, you can terminate the RMU Show Statistic operation by pressing Ctrl/C or Ctrl/Y and then typing "exit". When you are running the RMU Show Statistic utility in a batch job, terminate the operation by deleting the batch job.

The date and time specified by the UNTIL qualifier must represent an absolute time; that is, delta times such as "five minutes in the future" cannot be specified. However, on OpenVMS, DCL does allow the specification of delta times. The following example indicates that the RMU Show Statistic utility should stop in five hours:

```
$ RMU /Show Statistic /UNTIL=" ' f$cvtime("+5:00","ABSOLUTE") ' " -
  MF_PERSONNEL
```

The f\$cvtime() DCL lexical function also allows the specification of the keywords "TOMORROW", "TODAY" and "TODAY+n". For more information on the f\$cvtime() lexical function, please refer to the *VMS Run-Time Library* reference manual. Note that the "ABSOLUTE" keyword is required to generate an acceptable date format.

The UNTIL qualifier cannot be used if the INPUT qualifier is specified.

This qualifier cannot be modified at runtime.

There is no corresponding configuration variable name.

PRESENTATION
 FACILITIES

The RMU Show Statistic utility contains a number of presentation facilities. A facility is a specialized component that provides additional information or analytical capability. Each facility typically has its own sub-menu.

Not all facilities are always available. Some facilities depend on database attributes, such as global buffers being enabled. Others depend on the current state of the RMU Show Statistic utility, such as replaying a binary input file.

The following is a list of the available RMU Show Statistic utility facilities:

Facility	Description
Clusterwide Statistic Collection Facility	Allows the monitoring of statistics from any or all nodes in the cluster where the current database is open.
Configuration File Facility	Provides the ability to import and export the current utility settings and attributes.
Dashboard Facility	Allows the monitoring and temporary adjustment of various database attributes that affect database or process performance
Online Analysis Facility	Examines the runtime state of the database and reports inconsistencies or anomalies.
Process Monitoring Facility	Allows the monitoring of a specific process or group of processes.
Tools Facility	Collection of specialized tools and utilities.
User-defined Event Facility	Provides the ability to define events and their actions, based on the runtime statistic information.

Each of these facilities is discussed later in this handbook.

BINARY INPUT
 FILE REPLAY
 INFORMATION

The INPUT qualifier identifies the previously-recorded binary file from which you can read the statistics. This file must have been created by an earlier session of the RMU Show Statistic utility that specified the OUTPUT qualifier.

You cannot specify a database file name with the INPUT qualifier. Also, you must not use the CLUSTER, UNTIL, OUTPUT, or NOINTERACTIVE qualifiers with the INPUT qualifier. However, you can use the TIME qualifier to change the rate of the display. This will not change the computed times as recorded in the original session.

You can replay statistics gathered in a file by using the TIME qualifier.

For example, when you originally create the binary output file, you can use the OUTPUT qualifier to create a file of database statistics at a given refresh rate. For instance, you can record a session using the TIME=60 qualifier; this session will gather statistics once per minute.

You can also force frequent screen updates by using a negative number for the TIME qualifier. For example, TIME=-10 refreshes the screen every 10/100 (1/10) of a second. Note that you use a lot of system resources, particularly on the smaller CPU machines, when you specify this time interval.

You can then subsequently use the INPUT and TIME qualifiers to view the statistics again at a rate that you determine. For example, the following command will replay the PERS.LOG file and refresh the display once per second, thus replaying 10 hours of statistics in 10 minutes:

```
$ RMU SHOW STATISTIC /INPUT=PERS.LOG /TIME=1
```

If you do not specify the INPUT qualifier, you must specify the database-file-name parameter.

The Oracle Rdb 7.0 RMU Show Statistic utility has been enhanced to provide you with more control over the utility's display of database statistics in replay mode. When in replay mode, type "I" to select the "Input" option from the current on-screen menu. When you select the "Input" on-screen menu option, the "Select Input Control" menu for replay mode is displayed:

```
+-- Select Input Control --+
|
| A. Replay
| B. Pause
| C. Continue
| D. Place markpoint
|
+-----+

```

The following are the available menu options and their meanings:

Replay This option causes the input file to be replayed from the beginning. This can be selected from either replay mode or paused mode. When you select the "Replay" option, statistics are displayed from the beginning of the input file.

This option can be used in conjunction with the input file "Fast-forward" option, described later, to essentially perform a rewind operation.

Pause This option causes the reading of the binary input file to pause. When you select the "Pause" option, the RMU Show Statistic utility is in paused mode; note that the "Mode:" field in the display header changes from "Replay" to "Paused". The display remains in paused mode until you select the "Continue" option from the "Select Input Control" menu.

When you select the "Pause" option, the "Select Input Control" menu for paused mode is displayed. In this menu, option "B" is Single-Step:

```
+-- Select Input Control --+
|
| A. Replay
| B. Single-step
| C. Fast-forward
| D. Continue
| E. Place markpoint
+-----+
+-----+
```

Two new options (“Single-step” and “Fast-forward”) appear in the on-screen menu for the current display when pause mode is being used.

All the on-screen menu items for a display continue to operate normally in paused mode, except “Set_rate”. You can still change the display rate using the “Set_rate” on-screen menu option, but the statistics themselves are not changed.

In paused mode, for example, you can change the display format from numeric to graph format and back again. The “Options” and “Write” on-screen menu options are especially useful, because you know exactly what you are printing. You can also move between the available screens using the right angle bracket (>) and left angle bracket (<) keys when the display is paused. This allows you to acquire a true snapshot of a moment in time.

To advance by one record in the input file while in paused mode, you can select either option B (“Single-Step”) from the “Select Input Control” menu or type “S” to select the “Single-Step” option from the on-screen menu. When you are in paused mode, you can also select “Replay” mode (option “A” from the “Select Input Control” menu) and then step through all or part of the records in the input file using the “Single-Step” option.



Note that the operation of the “pause” option is different from the “Pause” on-screen menu option. The input file pause option stops the replay of the input file while the “Pause” on-screen menu option stops display of the statistic information.

Fast-forward⁴ This option allows you to search the binary input file, in a forward direction, for a specific date and time. This option is only available if you have previously paused the input file replay.

When this option is selected, you will be prompted to enter the date and time you wish to search for. The input file will be scanned for the first entry that occurred exactly at, or after the specified date and time. If the specified date and time has already expired, then the input file is not advanced.

Be careful when specifying just a time, as the date will default to the current date, not the date of the input file. This usually has the undesired effect of advancing the input file to the end.

⁴ Available in Oracle Rdb 7.0.2.0.

SWITCHING
 DATABASES

You can switch databases without exiting the RMU Show Statistic utility by using the Tools menu “Switch database [→“ option. The Tools menu is displayed when you type the exclamation point (!).

When this option is selected, you will be prompted to enter the name of the new database to be opened. As always, Ctrl/Z can be used to cancel the request. Also, as always, command recall is available to display previous terminal inputs.

Once you have entered a new database name, the utility will unbind from the current database, which may result in the database being closed, and bind to the new database. If the new database cannot be opened for any reason, the previously bound database will be re-bound. However, statistic collection will be restarted as of the time the database is first opened, which might be now if the database was opened.

The ability to switch to a new database is not available if you are replaying a binary input file (using the INPUT qualifier) or recording a binary output file (using the OUTPUT qualifier).

INDIVIDUAL
 PROCESS
 MONITORING ⁶

The RMU Show Statistic utility provides the capability to monitor individual processes, using the “Process Monitoring” facility. The purpose of this facility is to allow the monitoring of statistic information for specific processes or groups of processes.

The “Process Monitoring” facility is not available if clusterwide statistic collection is active. Conversely, the clusterwide statistic collection facility is not available when the “Process Monitoring” facility is active.

The “Process Monitoring” facility employs a two-step approach:

- The process to be monitored must have its global statistics collection “activated”. This is done once per process, or until the process is notified that it should deactivate global statistics collection.
- The RMU Show Statistic utility must “attach” to an activated process.

If the process to be monitored has not had its global statistics collection activated, then the RMU Show Statistic utility cannot attach to the process. Once a process has been activated, any subsequent RMU Show Statistic utility user on the same node can attach to that process.

Processes can be activated either implicitly by the database monitor, or explicitly by the RMU Show Statistic utility. The database monitor implicitly activates processes when the database is opened using the STATISTICS qualifier to the RMU Open command; this qualifier is explained later.



⁶ Available in Oracle Rdb 8.0.

When processes are implicitly activated by the database monitor, the “Mode:” section of the screen header region contains the keyword “Global” instead of the normal “Online”.



The RMU Show Statistic utility requires SYSGBL privilege in order to activate a process. This privilege is required even if the activation is implicit by the database monitor.

Note the subtle differences between being “activated” and “attached”. Activating a process also performs an implicit attach. You only need to attach to processes activating by other RMU Show Statistic utility users.

The RMU Show Statistic utility “Process Monitoring” facility contains two separate methodologies for monitoring processes:

- A specific process on the node can be selected for which all of the existing screens will be displayed using that process’ specific statistic information. When using this option, the header region “node:” designation is replaced with the selected PID.

This option is available when recording or replaying a binary output file.

- Any or all processes on a node can be selected (or de-selected) for which overview information for the selected processes can be displayed.

This option is not available when replaying a binary input file.

To activate the “Process Monitoring” facility, use the “Tools” sub-menu and choose the “Process monitoring” option. The Tools menu is displayed when you type the exclamation point (!). Selecting the “Process monitoring” option will display another sub-menu that contains the process monitoring options applicable to the current database state. For more information on the “Process Monitoring” facility options, please refer to Chapter 7.

The “Process Monitoring” facility provides several “overview” screens, located in the “Process Information” sub-menu. These overview screens provide process-to-process comparison information to identify specific processes that warrant additional investigation. The overview screens are not available when replaying a binary input file.

Processes for which global statistic collection is activated are suffixed with the “**G**” indicator. This notation is displayed on any screen that displays process identifiers.

Processes to which the RMU Show Statistic utility is attached are suffixed with the “**A**” indicator. This notation is displayed on any screen that displays process identifiers.

Note that the statistic information on the process overview screens cannot be reset. However, when monitoring a specific process, the statistic information can be reset.

You can use the “Zoom” on-screen menu option to obtain detailed information about the process being monitored, such as the user name and the image being executed.

In addition to the RMU Show Statistic utility enhancements, the RMU Open command has been enhanced. The STATISTICS qualifier has been enhanced to include the IMPORT and PROCESS_GLOBAL keywords.

The RMU Open command STATISTICS qualifier syntax is the following:

```
/[NO]STATISTICS [= ( [NO]IMPORT | [NO]PROCESS_GLOBAL ) ]
```

The semantics of the qualifier are the following:

<u>Qualifier</u>	<u>Semantics</u>
/NOSTATISTIC	importing of previously saved statistics is not performed, and process collection of global statistics is not activated
/STATISTIC	importing of previously saved statistics is performed, but process collection of global statistics is not activated
/STATISTIC=NOIMPORT	importing of previously saved statistics is not performed, and process collection of global statistics is not activated
/STATISTIC=IMPORT	importing of previously saved statistics is performed, and process collection of global statistics is not activated
/STATISTIC=PROCESS_GLOBAL	importing of previously saved statistics is performed, and process collection of global statistics is activated
/STATISTIC=NOPROCESS_GLOBAL	importing of previously saved statistics is performed, but process collection of global statistics is not activated

Oracle Corporation recommends that you do NOT use the RMU Open command STATISTICS=PROCESS_GLOBAL qualifier unless it is absolutely necessary for the analysis of performance problems. The STATISTICS=PROCESS_GLOBAL qualifier will cause each process attached to the database on that node to create a sizable global section, into which the process global statistic collection occurs. Oracle Corporation recommends that you activate process global statistic collection on a per-

process basis. That is, you should activate only those processes you are currently interested in, and deactivate them when you are with them.

The following formula defines the memory requirements for each process when it is activated, expressed in bytes:

$$\begin{aligned} &4096 + \\ &(512 * \text{"# storage areas"}) + \\ &((128 * \text{"# row caches"}) / 4) + \\ &(512 * \text{"max # logical areas"}) \end{aligned}$$

For example, the default MF_PERSONNEL.RDB requires approximately 300k (580 pages) of global section backing store memory, PER process attached by the utility. Obviously, this memory requirement increases as the size of the database increases.

Menus, Help and Prompts

Chapter 3 - Menus, Help and Prompts

To select the statistic screens that provide the many different statistics on database activity, or respond to prompts from the utility, you use the menus provided by the RMU Show Statistic utility. The menu options are quite dynamic, and vary depending on such factors as the current screen being displayed as well as the current database attributes that are active. For example, if your database does not have global buffers enabled, then the global buffer related menu options will not be displayed.

There are two basic forms of menus available: the “on-screen” menu and the “named” menu.

ON-SCREEN MENU

The RMU Show Statistic utility has a strip menu along the bottom of every screen; the options on the menu vary according to your current screen context. You simply type the highlighted letter that corresponds to the menu option you desire.

The following is an example of a on-screen menu, as it would appear on the bottom line of the screen.

```
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
-----
```

The following list identifies the complete set of on-screen menu options. Note that not all of the on-screen menu options are actually displayed in the menu, such as the “#”, “\$” and “D” options. Also, options may not be displayed if the screen does not contain adequate space. However, the options will still be available.



Note that some keystrokes in the list are ambiguous, such as the “**B**” key for both “Bell” and “Brief”. Each statistic screen is designed carefully to ensure that all on-screen menu options are unambiguous. For example, the “Bell” and “Brief” on-screen menu options will never appear on the same screen.

- Alarm This on-screen menu option allows you to control which stall messages will be displayed based on the duration of the stall.
- Bell This on-screen menu option rings the bell when the Alarm is tripped.
- Brief This on-screen menu option displays a brief report instead of a full report.
- Config This on-screen menu option displays the screen-specific configuration menu.
- Continue This on-screen menu option continues presenting statistics for a screen that is in single-step mode (see INPUT qualifier for more information).
 - D⁷ This on-screen menu option displays the main menu (obsolete; use the “Menu” option instead).
- Exit This on-screen menu option exits the RMU Show Statistic utility. Confirmation will be required if the OPTION=CONFIRM qualifier is specified.
- Filter This on-screen menu option allows you to filter screens containing text-based messages.
- Full This on-screen menu option displays a full report instead of a brief report.
- Graph This on-screen menu option displays a histogram graph instead of numbers.
- Help or ? This on-screen menu option displays the help menu.
 - Input This on-screen menu option displays the binary file controls submenu when the INPUT qualifier is used.
- LockID This on-screen menu option displays the “Lock Information” zoom screen. For more information on this on-screen menu option, please refer to the “Lock Information” section.
- Menu This on-screen menu option displays the main menu.
- Normal This on-screen menu option displays the normal screen contents. This option is used after selecting an alternate screen display format.
- Numbers This on-screen menu option displays the numeric statistics instead of graph.
- >next_page This on-screen menu option displays the next screen page.
- <prev_page This on-screen menu option displays the previous screen page.

⁷ Available in versions prior to Oracle Rdb 7.0 as “Display_menu” on-screen menu option.

- Options** This on-screen menu option displays a menu containing options for writing all of the screens to a text file named STATISTICS.RPT. This option should be used with care because it generates a very large output file.
- The “Start automatic screen capture” option of the Tools menu can be used to automatically execute the “Options” on-screen menu option at planned intervals.
- PageInfo** This on-screen menu option displays the “Database Page” zoom screen. For more information on this on-screen menu option, please refer to the “Page Information” section.
- Pause** This on-screen menu option pauses statistic collection (numeric statistics only). Selecting the Pause option again resumes statistic collection.
- Note that this on-screen option differs from the “Input” on-screen menu “pause” option.
- Refresh** This on-screen menu option refreshes clusterwide database information.
- Reset** This on-screen menu option resets the statistics.
- Set_rate** This on-screen menu option sets the display update rate.
- Step** This on-screen menu option single-steps through a binary input file. This on-screen menu option is only available when the INPUT qualifier is specified.
- Time_plot** This on-screen menu option plots a specific field's value by time.
- Unreset** This on-screen menu option restores the original statistic values after using the “Reset” option. This on-screen menu option is only available when the “Reset” on-screen menu has been previously selected.
- Update** This on-screen menu option updates the database dashboard screen information.
- Write** This on-screen menu option writes the current screen (or zoom screen) to a text file named RMU.SCR.
- X_plot** This on-screen menu option plots a specific field's value using scatter-based display.
- Yank** This on-screen menu option yanks and puts a specific field from the current screen onto the “Custom Statistics” screen.
- Zoom** This on-screen menu option displays a zoom screen.
- ! This on-screen menu option displays the Tools menu.
- # This on-screen menu option displays the current menu. This on-screen menu option is never displayed on the screen, but is always available.

- § This on-screen menu option invokes an operating system command. This on-screen menu option is never displayed on the screen, but is always available. For more information on invoking an operating system command, please refer to the “Invoke command” section.
- @⁸ This on-screen menu option invokes the OpenVMS mail system to send or receive mail. This on-screen menu option is never displayed on the screen, but is always available. For more information on sending or receiving email, please refer to the “Send/Receive mail” section.

NAMED MENUS

A named menu is a standard box-style menu, and it is identified with a name or description on the border. Every menu option is prefixed with either a letter [A-Z] or a number [0-9]. This is known as the shortcut keystroke. You either type the letter or number that corresponds to the menu option you desire, or change the highlighted menu option.

For example:

```
+--- Online Help -----+
| A. Using the keyboard   |
| B. Screen description   |
| C. Field descriptions    [->|
| D. Search for help on... [->|
| E. Goto screen "by name" |
| F. Goto screen          [->|
+-----+

```

The notation “[→” indicates that the option contains a submenu of more options.

The following table shows the universal methods available to navigate through the various named menus:

<u>Shortcut Key-stroke</u>	<u>Description</u>
Help or PF2 or ?	Displays the help menu.
Up-arrow (↑) or down-arrow (↓)	Highlights the menu option above (or below) the currently highlighted option.
Left-arrow (←) or	Highlights the menu option to the left (or right) of the currently highlighted option; this keystroke can only be used in multi-

⁸ Available in Oracle Rdb 7.0.2.0.

<u>Shortcut Key-stroke</u>	<u>Description</u>
Right-arrow (⇒)	column menus.
Return or Enter or Select	Selects the highlighted menu option.
Ctrl/W	Repaints the screen.
Ctrl/Z	Cancels a menu or exits RMU Show Statistic utility. To prevent accidentally exiting the utility, use the OPTION=CONFIRM qualifier.
A-Z or 0-9	Selects the corresponding menu option.

Duplicate Menu Options⁹ Occasionally, a named menu may contain more than 36 entries. When a named menu contains more than 36 entries, some of the shortcut keystroke options may be repeated. For example, on a menu containing 37 entries, the menu will contain two shortcut keystroke “A.” options. If you choose a duplicate menu option, you will be presented with a submenu that contains all of the duplicate entries; you may then refine your selection using the submenu.

If you select a duplicate menu option using the “Return” or “Enter” key, then the selected option is used. You will not be presented with a submenu of the other options containing the same shortcut keystroke.

Logical Area Menu Filters A logical area is a table or an index of a database. The “Logical Area” named menu can be difficult to migrate when production databases contain hundreds or thousands of tables and indices. Therefore, the contents of the “Logical Area” menu can be controlled by the use of wildcard selection criteria, known as a filter.

The Tools menu, obtained using the exclamation point (!) from any screen, contains the “Logical area menu filter” option. This option lets you specify, or delete, a search pattern containing wildcards.

The specified pattern must match at least one logical area, or the pattern will be rejected.

The filtered logical area menu is only available when displaying all logical areas. It is not available if you selected the “Display Application Logical Areas” option from the Tools menu.

⁹ Available in Oracle Rdb 7.0.2

The specified pattern string may contain either one or both of the two wildcard characters, an asterisk (*) or a percent (%). The asterisk character is mapped to zero or more characters. The percent character is mapped to only one character. For example, the pattern “*EMP*” will find any logical area containing the text “EMP”, while the pattern “EMP%” will find only those logical areas whose name starts with “EMP”.

MINI-MENUS

Mini-menus are often used to request additional user information. These types of menus are most often used for zoom screens, but also when presenting the user with a list of selectable choices. Most mini-menus do not contain a border and simply overlay the screen region.

RMU SHOW
 STATISTIC MENU
 OVERVIEW

To display the “Main” menu, select the “Menu” on-screen menu option.

The following is the complete list of submenus available in the main menu for the RMU Show Statistic utility, in alphabetical order:

<u>Menu Name</u>	<u>Description</u>	<u>Restrictions</u>
Database Dashboard	Run-time database attribute information, including global and per-process information	Not available if INPUT qualifier specified
Database Parameter Info	Database and operating system attribute information, similar to that available using the RMU Dump Header utility	Not available if INPUT qualifier specified
General Information	General (obscure) information	None
Global Buffer Information	Global Buffer information	Not available if global buffers disabled or if INPUT qualifier specified
Help	Online help	“?” or PF2 keystroke
Hot Standby Information	Hot Standby feature information	None
Index Information	Index summary information	None
IO Statistics (by file)	I/O information on a per-storage area basis, including storage area overview information	Not available if INPUT qualifier specified unless binary file created with OUTPUT=area qualifier

<u>Menu Name</u>	<u>Description</u>	<u>Restrictions</u>
IO Statistics Information	IO summary information	None
Journaling Information	AIJ and RUJ file menu. Also includes general process failure recovery screens	None
Lock Statistics (by file)	Lock information on a per-storage area basis, including storage area overview information	Not available if INPUT qualifier specified unless binary file created with OUTPUT=area qualifier
Locking (one lock type)	Lock information for a specific lock type, including summary lock information	None
Locking (one stat field)	Lock information for a specific statistic field	None
Logical Area Information	Logical Area (tables and indices) information	Not available if NOLOGICAL_AREA qualifier specified, or INPUT qualifier specified
Main Menu	Main menu displayed when "Menu" on-screen menu option selected comprised primarily of summary statistic information screens	None
Objects (one stat field)	Database rootfile "object" information for a specific statistic type	None
Objects (one stat type)	Database rootfile "object" information for a specific object type	None
Online Analysis & Info.	Online database and application statistical analysis	Not available if INPUT qualifier specified
Process Information	Application process information menu	Not available if INPUT qualifier specified
Row Cache (One Cache)	Row cache information for a specific row cache	Not available if row cache disabled or INPUT quali-

<u>Menu Name</u>	<u>Description</u>	<u>Restrictions</u>
		fier specified unless binary file created with OUTPUT=row_cache qualifier
Row Cache (One Field)	Row cache information for a specific statistic type	Not available if row cache disabled or INPUT qualifier specified unless binary file created with OUTPUT=row_cache qualifier
Row Cache Information	Detailed row cache information, displayed for a specific row cache	Not available if row cache disabled or INPUT qualifier specified unless binary file created with OUTPUT=row_cache qualifier
Tools	Collection of tools and RMU Show Statistic utility processing options	exclamation point (!)

Online Help

Online help is available from all screens and most menus. You can select the “Help” facility at any time using the PF2 or “?” keys, and for screen displays you can also use the “H” key.

Note that when a named menu is displayed, pressing “H” will select the menu option identified by the “H.” prefix; you must use the PF2 or “?” keys when named menus are displayed.

The following is an example of the RMU Show Statistic help menu:

```

+-----+
| A. Using the keyboard           |
| B. Screen descriptions          |
| C. Field descriptions           | [->
| D. Search for help on...       | [->
| E. Goto screen "by name"      |
| F. Goto screen                 | [->
+-----+
    
```

HELP SCREEN
 NAVIGATION

In some cases, the help text is quite extensive and may require more than one page. The size of the help screen is based on the number of lines configured for the terminal; for example, a workstation window configured to display 48 lines will have a larger help screen than a VT100 terminal configured to display 24 lines.

When the help screen contains additional unseen information, the indicators “MORE→” or “←MORE” will be displayed at the top or bottom of the help screen, identifying which direction contains more help information.

You can scroll through the multiple pages of a help screen, either one line at a time, or one page at a time.

The following table shows the universal methods available to navigate through the various pages of a help screen:

<u>Keystroke</u>	<u>Description</u>
Up-arrow (↑) or down-arrow (↓)	Displays the previous or next line of the help screen
Left-arrow (←) or Right-arrow (→)	Displays the previous or next page of the help screen.
W	Writes the current screen to a text file named RMU.SCR.

<u>Keystroke</u>	<u>Description</u>
Ctrl/W	Repaints the help screen.
Ctrl/Z	Cancels the help screen.

Note that you cannot display help about the “Help” facility.

PROCESS
 IDENTIFIER HELP ¹⁰

At any prompt requesting you to enter a process identifier, typing “?” will display a menu of attached processes on the node. This is an easy way to obtain a process identifier if you are displaying a screen that does not contain any process identifier information.

TOOLS AND
 PREFERENCES
 MENU

The RMU Show Statistic utility contains a collection of general-purpose tools and preferences. The Tools menu is available from all screens using the exclamation point (!).

For additional information on the Tools menu, please refer to Volume 1, Chapter 7.

User Prompts

Often, the RMU Show Statistic utility will need to prompt you to enter additional information. These are always requests for dynamic information where a menu would not suffice. For example, when changing the screen refresh rate, you will be prompted to enter the desired number of seconds. Since you may desire to specify any refresh rate, it is not appropriate to provide a menu containing all valid rates.

USER RESPONSE
 REGION

All user responses are entered in the on-screen menu region of the screen; this is the last line of the screen. Your response length is limited by the size of the display; for example, a VT100 terminal configured with a screen width of 80 characters can accept at most 80 characters for your response.

Command recall is available to help simplify your responses to repetitive prompts.

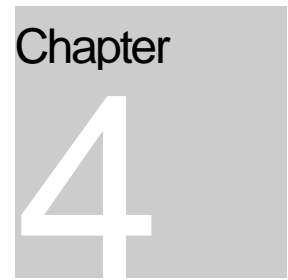
¹⁰ Available in Oracle Rdb 7.0.2.

USER RESPONSE
TIMEOUT

By default, all prompts will time out if you do not respond to them in adequate time. The default user response timeout interval is 60 seconds.

The default user response timeout interval can be specified using the `PROMPT_TIMEOUT=interval` qualifier, or the `PROMPT_TIMEOUT=interval` configuration variable. The user response timeout can also be modified at runtime using the Tools menu “set user-prompt timeout” option.

Note that the user response timeout interval also affects the online help screens and all zoom screens.



Presentation Screens

Chapter 4 - Presentation Screens

This chapter provides information about the various methods with which the RMU Show Statistic utility presents statistic information.

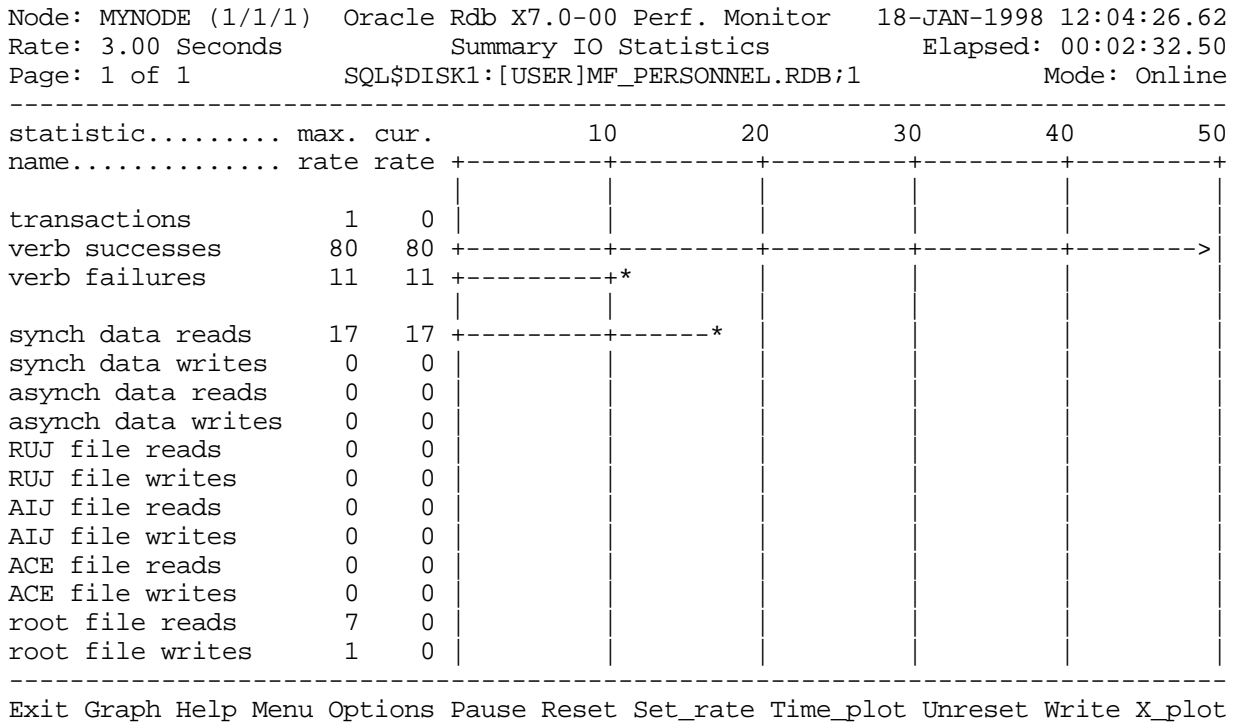
SCREEN DISPLAY

The basic presentation medium of the RMU Show Statistic utility is the screen. Although the CLI-based utility is limited to displaying only one statistic screen on your terminal or workstation at any time, all of the statistic information is constantly maintained, including the averages and rate information.

SCREEN SIZE

The screen size is based on the number of lines configured for your workstation or terminal. For example, a workstation window configured for 48 lines will display a larger screen than a VT100 terminal configured for 24 lines. However, *most* columnar (numeric) statistic screens are designed to fit on a 24-line terminal.

The following is an example of a typical graphical (histogram) screen:



The following is an example of the same screen, in columnar (numeric) format:

```

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds Summary IO Statistics Elapsed: 05:39:13.29
Page: 1 of 1 SQL$DISK1:[USER]MF_PERSONNEL.RDB;1 Mode: Online
-----
statistic..... rate.per.second..... total... average..
name..... max. cur. avg... count... per.trans
transactions      0   0   0.0      2      1.0
verb successes    2   0   1.7     250    125.0
verb failures     0   0   0.2     32     16.0
synch data reads  0   0   0.4     66     33.0
synch data writes 0   0   0.0      0      0.0
asynch data reads 0   0   0.0      0      0.0
asynch data writes 0   0   0.0      0      0.0
RUJ file reads    0   0   0.0      0      0.0
RUJ file writes   0   0   0.0      0      0.0
AIJ file reads    0   0   0.0      0      0.0
AIJ file writes   0   0   0.0      0      0.0
ACE file reads    0   0   0.0      0      0.0
ACE file writes   0   0   0.0      0      0.0
root file reads   0   0   0.1     20     10.0
root file writes  0   0   0.0      5      2.5
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot

```

SCREEN PAGES

Some tabular screens contain more information than will fit on the corresponding terminal window size. When this occurs, multiple pages of information are available for the screen. The number of pages for a given screen is displayed in the screen's header region.

You migrate to these pages using the right-angle bracket (>) or left-angle bracket (<) keystrokes, as well as several other methods described in the Screen Migration section later in this chapter.

Columnar statistic screens typically have only one page of information. That is, the numeric displays are designed to be displayed on a terminal whose window size is 24 lines.

Screen Regions

Every RMU Show Statistic utility screen contains three regions: header, body and status regions.

HEADER REGION

The first region of a screen is the header region. The header provides information about the state of the utility as well as the current page. The header region is always three lines long, and each line contains a left segment, center segment and right segment.

The following is an example of a typical header region:

```
Node: MYNODE (1/1/2)  Oracle Rdb V7.0-00 Perf. Monitor 15-JAN-1998 13:10:27.90
Rate: 1.00 Second      Summary IO Statistics                Elapsed: 00:09:13.89
Page: 1 of 1  DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1  Mode: Online
```

The first line of the header region contains the following information:

<u>Header Region Segment</u>	<u>Description</u>
Node: MYNODE (1/1/2)	Statistic collection node information; see description below
Oracle Rdb V7.0-00 Perf. Monitor	Oracle Rdb product version number
15-JAN-1998 13:10:27.90	Current date and time

The second line of the header region contains the following information:

Header Region Segment	Description
Rate: 1.00 Second	Screen refresh rate, in seconds
Summary IO Statistics	Current screen name
Elapsed: 00:09:13.89	Elapsed time since utility started or last "Reset". Note that if the STATISTICS ¹¹ qualifier was specified on the RMU Open command, this is the elapsed time since statistic information was originally started.

The third line of the header region contains the following information:

Header Region Segment	Description
Page: 1 of 1	Current screen page number and maximum number of pages
DISK\$: [DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1	Current database name
Mode: Online	Operational mode. Modes are "Online", "Record", "Replay" and "Paused".

COLLECTION NODE
INFORMATION

The Oracle Rdb 7.0 version RMU Show Statistic utility screen header region includes enhancements to display the current node suffixed with a "(x/y/z)" identifier. The "x" value indicates the number of nodes on which statistics are being actively collected. The "y" value indicates the number of nodes on which the database is being accessed. The "z" value indicates the maximum number of nodes on which the database is accessible.

For example, the indicator "(1/3/5)" specifies that statistics are being collected from 1 node, but that the database is currently open on 3 nodes, with a maximum open node count of 5.

The "Node" tag changes to "Cluster" when statistic collection occurs on more than one node. This indicates that clusterwide statistic collection is active.

When using the "Process Monitoring" facility to monitor a specific process, the "Node" tag changes to the process identifier currently being monitored.

¹¹ Available in Oracle Rdb 7.0.2.

BODY REGION

The body region of the screen is where the actual statistic information is presented.

The following is an example of a typical numeric screen body:

```

-----
statistic.....      rate.per.second.....      total...      average..
name.....           max.      cur.      avg...      count...      per.trans
transactions         0         0         0.0         2             1.0
verb successes       2         0         1.7        250           125.0
verb failures        0         0         0.2         32            16.0
synch data reads    0         0         0.4         66            33.0
synch data writes   0         0         0.0         0             0.0
asynch data reads   0         0         0.0         0             0.0
asynch data writes  0         0         0.0         0             0.0
RUJ file reads      0         0         0.0         0             0.0
RUJ file writes     0         0         0.0         0             0.0
AIJ file reads      0         0         0.0         0             0.0
AIJ file writes     0         0         0.0         0             0.0
ACE file reads      0         0         0.0         0             0.0
ACE file writes     0         0         0.0         0             0.0
root file reads     0         0         0.1         20            10.0
root file writes    0         0         0.0         5             2.5
-----
    
```

Statistic information in the body region of a screen can be displayed in the following formats:

Display Format	Description
Numeric	Displays numeric statistic information in columnar format.
Columnar (histogram/graph)	Displays occurrence-per-second rate information in a histogram chart format.
Tabular (numeric)	Displays numeric statistic information in tabular format.
Tabular (text)	Displays textual information in a tabular format.
Time-based Plot	Displays occurrence-per-second rate information in a time-based chart format.
Scatter-based Plot	Displays occurrence-per-second rate information in a scatter-based chart format.
Dashboard (numeric)	Displays database dashboard runtime information in a numeric tabular format.

STATUS REGION

The third region of the screen is the status bar. Typically, the on-screen menu is displayed in the region, but the region also serves several other purposes:

On-screen Menu The on-screen menu is normally displayed in the status region. The on-screen menu lets you quickly take screen-specific actions with a single keystroke.

The following is an example of typical on-screen menu:

```
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot
-----
```

Menus, in general, are explained in Chapter 3.

Error Messages Any error messages encountered are displayed in this region. These error messages are automatically erased after 10 seconds, unless you have disabled automatic screen refreshes in the Tools menu.

Broadcast Messages Operating system broadcast messages are displayed on the status region, unless you have specified the NOBROADCAST qualifier.

User Response Region The status region is also where user responses to prompts are entered. Command recall is available to help simplify your responses to repetitive prompts.

Screen Body Display Formats

The screen body region provides six different display formats: graph, numeric, numeric tabular, tabular, time plot, and scatter plot.

GRAPH DISPLAY
 FORMAT

The graph display format displays realtime occurrence-per-second for each statistic, in a histogram presentation. The bars of the graph reflect current statistic rates; that is, the occurrences-per-second for a particular statistic during the last sample interval.

You use the “Graph” on-screen menu option to select the graph display mode. You use the “Number” on-screen menu option to switch from graph display mode to numeric display mode.

The following is an example of the graphical display format of the screen body region:

statistic.....	max.	cur.	10	20	30	40	50
name.....	rate	rate					
transactions	1	0					
verb successes	80	80	----->				
verb failures	11	11	-----*				
synch data reads	17	17	-----*				
synch data writes	0	0					
asynch data reads	0	0					
asynch data writes	0	0					
RUJ file reads	0	0					
RUJ file writes	0	0					
AIJ file reads	0	0					
AIJ file writes	0	0					
ACE file reads	0	0					
ACE file writes	0	0					
root file reads	7	0					
root file writes	1	0					

The layout of the graph format is explained in the following list:

- statistic name This column identifies the individual statistic.
- max rate This column indicates the maximum occurrence-per-second rate of the statistic since the RMU Show Statistic utility was started or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- cur rate This column indicates the current occurrence-per-second rate of the statistic during the last sample interval (screen refresh); you can change the sample interval using the “Set_rate” on-screen menu option.
- Out-of-Range Indicator The bars on the chart reflect the current rate, but are limited by the width of the terminal display.

The size of the graph chart is determined by the terminal width. For terminals with a display width of 80 columns, the chart displays current rate values up to 50. For terminals with a display width of 132 columns, the chart displays current rate values up to 100. The formula for determining the maximum current rate of the graph chart is the following: “terminal_width - 30”.

For example, a terminal with a display width of 80 columns would display a current rate up to (80-30) or 50.

For current rates less than the maximum, the graph bar ends with an asterisk (*). For current rates greater than the maximum, the bar ends with a right angle bracket (>); the actual value is shown in the current rate column.

Because of the relatively small current rate value information that can be displayed, especially with the current high-speed hardware technology available, Oracle does not recommend using the graph chart for detailed performance analysis. However, it is useful for identifying spikes in a graphical manner.

NUMERIC DISPLAY
FORMAT

The numeric display format presents statistic information in a columnar chart. The chart provides more detail than the graph format, such as rates, averages and total values.

You use the “Numbers” on-screen menu option to select the numeric display mode. You use the “Graph” on-screen menu option to switch from numeric display mode to graph display mode.

The following is an example of the numeric display format of the screen body region:

```

-----
statistic.....      rate.per.second.....      total...      average..
name.....           max.      cur.      avg...      count...      per.trans
transactions         0         0         0.0         2             1.0
verb successes       2         0         1.7        250           125.0
verb failures        0         0         0.2         32            16.0
synch data reads    0         0         0.4         66            33.0
synch data writes   0         0         0.0          0             0.0
asynch data reads   0         0         0.0          0             0.0
asynch data writes  0         0         0.0          0             0.0
RUJ file reads      0         0         0.0          0             0.0
RUJ file writes     0         0         0.0          0             0.0
AIJ file reads      0         0         0.0          0             0.0
AIJ file writes     0         0         0.0          0             0.0
ACE file reads      0         0         0.0          0             0.0
ACE file writes     0         0         0.0          0             0.0
root file reads     0         0         0.1         20            10.0
root file writes    0         0         0.0          5             2.5
-----

```

The layout of the numeric format is explained in the following list:

- statistic name Identifies the individual statistic.
- rate per second (max) This column indicates the maximum occurrence-per-second rate of the statistic since the RMU Show Statistic utility was started or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- rate per second (cur) This column indicates the current occurrence-per-second rate of the statistic during the last sample interval (screen refresh); you can change the sample interval using the “Set_rate” on-screen menu option.

- rate per second (avg) This column indicates the average occurrence-per-second rate of the statistic since the database was opened or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- total count This column indicates the total number of occurrences of the statistic since the database was opened or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- average per trans This column indicates the transaction average of the total-count column and the total number of completed transactions.

NUMERIC TABULAR
DISPLAY FORMAT

The numeric tabular display format presents statistic information in a mixed-format columnar chart. This type of display is available for use for the “File IO Statistics” screens only, as it provides a comparison of occurrences-per-second with throughput-per-second.

The numeric tabular display does not have the “Graph” and “Numbers” on-screen menu options. It is a specific-purpose display for storage area statistic information.

The following is an example of the numeric tabular display format of the screen body region:

```

-----
                          For File: Database Root
-----
statistic..... rate.per.second..... total..... a verage.....
name..... max..... cur..... avg..... count..... per.trans....

total I/O rates                                1                35845           1.1
(Synch. reads)                0                0                0.0             26             0.0
(Synch. writes)               5                0                0.8            6212           0.2
(Extends)                     0                0                0.0             0              0.0
(Asynch. reads)               0                0                0.0             0              0.0
(Asynch. writes)             16                1                3.9            29607          0.9

statistic..... blocks.transferred..... stall.time.(x100).....
name..... avg.per.I/O.. total..... avg.per.I/O... total.....

total I/O sizes                1.0              36004           0.6            22477
(Synch. reads)                1.8              48              0.3             9
(Synch. writes)               1.0              6221           3.3            20914
(Extends)                     0.0              0              0.0             0
(Asynch. reads)               0.0              0              0.0             0
(Asynch. writes)             1.0              29735          0.0            1554
-----

```

The layout of the numeric tabular format is described below:

- For File This field identifies the specific storage area to which the statistic information applies.
- rate per second This section provides information on the rate at which I/O operations are performed by the database. A single I/O operation may access several blocks of information; thus, the number of I/O operations do not correlate directly to the volume of data manipulated.

However, if the file is on a disk by itself and this number is high (greater than 30 for RA8n disks), the file may be a good candidate for partitioning. This number also indicates how much use the file is getting, and if the file should be on its own disk.

The information is partitioned into the following categories:

- max. This category indicates the maximum number of operations per second attained for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations. The “max.” category is typically used to differentiate peaks in performance from the system-wide average. Note that no total figure is displayed for this category.
- cur. This category indicates the current number of operations per second attained for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations. This information is computed using the number of I/O operations performed during the screen-update time interval.
- avg. . . This category indicates the average number of operations per second attained for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations. The avg. category is typically used to determine overall system activity. Note that no total figure is displayed for this category.
- total count This category indicates the total number of synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations performed by the database. The total count category serves as the basis for the “Average Per I/O” categories. It is a good indicator of how much I/O activity there is for a file and for comparing one file's activity with other files in the database.
- average per trans This category indicates the total number of synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations, based on the total number of application transactions committed by all processes attached to the database. The total number of committed transactions is not displayed. The processing within a transaction and the duration of a transaction for this category are determined entirely by the application.

The “average per trans” category can indicate whether or not certain transactions incur a high number of I/O operations to accomplish a task. If you notice a number that is higher than you expected, you should investigate further by using other RMU Show Statistic utility options.

- blocks transferred This section provides information on the volume of data manipulated (transferred) by the database. Depending on the operation, Oracle Rdb attempts to make an I/O operation do as much work as possible by transferring as many blocks as possible. The blocks transferred statistic shows how much work is being done for each I/O operation. A number less than the buffer size indicates that either SPAM pages are used heavily (because they fill only one page), or there are contention problems because Oracle Rdb resolves contention problems by writing and reading pages rather than full buffers.

The information is partitioned into the following categories:

- avg. per I/O This category indicates the average number of blocks transferred for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write opera-

tions, based on the total number of I/O operations that have been performed by the database.

- total . . . This category indicates the total number of blocks transferred for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations.
- stall time (x100) This section provides information on how long processes wait for the I/O operations to complete. Processes typically stall due to contention for the disk by other processes. Stall time is measured in hundredths of a second but is displayed as whole numbers. Thus, a displayed value of 100 represents 1 second of stall time. A large number can indicate excessive I/O activity to a file and that requests are becoming stacked in a queue.

This indicates the file is a good candidate for partitioning or being placed on another disk.

Normally, RA8n disks should complete an I/O operation in 2/100 to 4/100 of a second, while RD5n disks should complete an I/O operation in 5/100 to 8/100 of a second.

The information is partitioned into the following categories:

- avg. per I/O This category indicates the average time stalled for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations based on the number of I/O operations that have been performed.

When you are displaying statistics for all data/snap files, the value for the average stall time per I/O field may not be what you expect if you have displayed the average stall time per I/O for the individual storage area files (.RDA) and snapshot files (.SNP) in the database. For example, it is possible for the average stall time per I/O for the all data/snap files screen to be 1.1 when the average stall times per I/O for the storage area files (.RDA) and snapshot files (.SNP) files are 8.3, 5.7, 6.4, and 6.8, respectively.

Oracle Rdb issues write I/O operations in parallel, asynchronously (this is a batch-write mechanism). This means that the average stall time per I/O for the all data/snap files screen is not the “average of the averages” shown for the individual storage area files (.RDA) and snapshot files (.SNP) files; this would imply that the I/Os completed serially. Rather, the total I/O stall time is “the average of the averages divided by the number of I/Os.” The average for the all data and snap files screen is usually a fraction of the individual file averages because the stall time is amortized across all I/Os issued in parallel.

For example, assume that Oracle Rdb does a batch-write operation (all in parallel, of course) to four storage areas. Assume that each individual storage area's I/O operation takes 20 milliseconds. If the I/Os were done serially, then the average stall time for all storage areas would be 20 milliseconds. However, because the I/Os are done in parallel, the average for all areas is actually 5 milliseconds (20 milliseconds divided by 4 I/O operations).

total. . . This category indicates the total time stalled for synchronous read, synchronous write, extend, asynchronous read, and asynchronous write operations.

TIME PLOT
 DISPLAY FORMAT

The time plot display format provides a detailed graph of the occurrences-per-second for a particular statistic, as they occur over time.

You use the “Time_plot” on-screen menu option to select the time plot display mode. You use the “Normal” on-screen menu option to switch from time plot display mode back to either numeric or graph display mode, whichever was previously active.

If the on-screen menu lists Time_plot, you can type “T” to request a time plot. After you type “T”, select a statistic from the screen by typing the letter associated with the desired field, or by using the arrow keys to move the cursor to the desired field and then pressing the Return key.

The following is an example of the time plot display format of the screen body region:

```

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 1.00 Second Summary IO Statistics Elapsed: 00:00:19.57
Page: 1 of 1 DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online
-----
                                transactions
Maximum rate = 0 Current rate = 0 Average rate = 0.0
Total count = 0 Average per transaction = 0.0
+-----+-----+-----+-----+-----+-----+
46+      *^^^*^^^*^^^*^^^*
41-45    *
36-40   **
31-35   *
26-30   *
21-25
16-20
11-15
6-10
1-5
+-----+-----+-----+-----+-----+-----+
                                Sample interval is 1.00 Second
-----
Exit Help Menu Normal Pause Reset Set_rate Write !
  
```

The layout of the time plot display format is explained in the following list:

- statistic name Identifies the individual statistic (“transactions” in this example).
- maximum rate This column indicates the maximum occurrence-per-second rate of the statistic since the RMU Show Statistic utility was started or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- current rate This column indicates the current occurrence-per-second rate of the statistic during the last sample interval (screen refresh); you can change the sample interval using the “Set_rate” on-screen menu option.

- total count** This column indicates the total number of occurrences of the statistic since the database was opened or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
- average per transaction** This column indicates the transaction average of the total-count column and the total number of completed transactions.
- Out-of-Range Indicator** The circumflex characters (^) in the top row indicate that the occurrences-per-second rate for the statistic is larger than the highest range of the chart. To change the chart range, use the “Reset” on-screen menu option; this option will automatically re-display the screen with a new set of ranges.

The Tools menu provides an option that allows you to enable or disable the Time_plot baseline. Enabling the baseline updates the graph continuously.

SCATTER PLOT
DISPLAY FORMAT

The Scatter Plot display format, similar to the Time Plot display format, shows information for a selected statistic field on the current screen. However, the information in the scatter plot is displayed in a vertical histogram. The Scatter Plot display format allows you to determine the composition of the values that determine the average rate statistic.

If the on-screen menu lists X_plot, you can type “X” to request a scatter plot. After you type “X”, select a statistic from the screen by typing the letter associated with the desired field, or by using the arrow keys to move the cursor to the desired field and then pressing the Return key.

After you have selected a field, the Scatter Plot display initially shows that the scale of the display is based on the running average rate for the selected field. This scale can be changed using the Reset menu option.

Each vertical line of the screen identifies the current occurrence-per-second rate bucket based on the selected statistic rate. The complete set of buckets is known as the collection range and comprises the entire Scatter Plot screen. For example, the Scatter Plot screen may have a range from 0.0 to 1.5 seconds, with each bucket representing 0.1 (one-tenth) of a second.

Based on the selected screen refresh rate, the normalized current rate for the selected statistic field is examined to determine into which bucket the rate would occur. This algorithm results in a current rate scattered distribution graph, which is extremely beneficial for performance analysis.

You use the “X_plot” on-screen menu option to select the scatter plot display mode. You use the “Normal” on-screen menu option to switch from scatter plot display mode back to either numeric or graph display mode, whichever was previously active.

The following is an example of the scatter plot display format of the screen body region:

```

Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 09:36:24.03
Rate: 1.00 Second       PIO Statistics--SPAM Access           Elapsed: 01:24:22.85
Page: 1 of 1           DISK$: [DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
Out-of-range count:    0.3--- = 0                               6.0+++ = 0
Current statistic range:  low = 0.5                             high = 1.9
                                fetch for read
Scaled distribution of statistic rate (in seconds)              Elapsed: 00:01:15.48
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
|         |         |         |         |         |         |         |         |         |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
0.0  0.4  0.8  1.2  1.6  2.0  2.4  2.8  3.2  3.6  4.0  4.4  4.8  5.2  5.6  6.0+
      (Each "*" represents 2 statistics updates) [A=Avg, M=Mean, R=RunAvg]
-----
Config Exit Help Menu Normal Pause Reset Set_rate Write !
  
```

The layout of the scatter plot display format is explained in the following list:

- out-of-range count: 0--- Identifies the number of statistics updates that are below the minimum recorded rate
- out-of-range count: 6.0+++ Identifies the number of statistics updates that are above the minimum recorded rate
- current statistic range: low = Identifies the lowest recorded rate, which is available for use as the basis for rescaling the display when you select the “Reset” on-screen menu option. This is also information about the boundary data conditions
- current statistic range: high = Identifies the highest recorded rate, which is available for use as the basis for rescaling the display when you select the “Reset” on-screen menu option. This is also information about the boundary data conditions
- statistic name Identifies the individual statistic (“fetch for read” in this example).
- elapsed: Indicates the elapsed time during which this scatter plot was generated.
- Sliding Indicators Three sliding indicators are displayed along the bottom horizontal axis:

Indicator	Description
R	The letter R indicates the running average rate, which is computed over the total elapsed time as shown in the summary region. Note that the R running average rate may not always appear, depending on the collec-

<u>Indicator</u>	<u>Description</u>
	tion range. Also, resetting the elapsed time is useful for comparing the running average against the collected average.
A	The letter A indicates the collected average rate, which is computed on the collection time.
M	The letter M indicates the collected median rate, which is also computed on the collection time.

Collection Buckets The number of collection buckets is based on the display width of the terminal. By default, each bucket comprises 5 columns of the display; the number of columns can be configured by the user. For example, setting the terminal width to 132 columns allows you to display more information than with an 80-column terminal width. Decreasing the number of columns per bucket achieves the same effect on an 80-column terminal width, but with a corresponding loss of precision.

Notice that the collected median is significantly higher than the collected average. This is indicative of skewed data, typically caused by periodic blips in the runtime performance. Identifying and eliminating these blips produces a smoother-running system.

Notice also that these occasional blips tend to get lost as noise in the running average. However, removing these blips ultimately improves the overall running average of the system.

Ideally, the “Scatter Plot” screen should resemble the classical bell-shaped curve data distribution.

The “Reset” on-screen menu option, instead of resetting the statistics values, resets the “Scatter Plot” screen scale based on the lowest and highest rate values actually recorded. This on-screen menu option is very useful for focusing the display on the actually occurring statistics update rates.

The “Set_rate” on-screen menu option causes the Scatter Plot information to be cleared and data collection restarted. This is required because the new rate's data collection patterns are unrelated to the old rate's data collection patterns.

The “Config” on-screen menu contains options that allow you to fine-tune the rate collection parameters and the display appearance. Currently, you can perform two configuration operations: set the number of displayed columns per bucket, and set the rate collection range. Both of these settings allow you more control over what data is collected and displayed.

Decreasing the number of columns per bucket allows you to collect more rates. Conversely, increasing the number of columns per bucket allows you to collect fewer rates,

but with more accurate rate reporting (that is, better scaling) per bucket. The minimum number of columns is 2 and the maximum is 10.

Setting specific collection range values allows you to determine the actual collection range, which is normally based on the rates already collected.

TABULAR DISPLAY
 FORMAT

The tabular display format is available for use primarily for textual or table-based information. In most cases, the screen display appears as a table of information.

The following is an example of the tabular display format of the screen body region:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 19-JAN-1998 12:36:54.65
Rate: 1.00 Second          Defined Logicals          Elapsed: 00:00:33.11
Page: 1 of 5  DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1  Mode: Online
```

```
-----
```

Logical.Name.....	Table.Name.....	Logical.Definition.....
RDM\$BIND_ABS_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BIND_ALS_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BIND_BUFFERS	LNMS\$PROCESS_TABLE	500
RDM\$BIND_DBR_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BIND_LCS_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BIND_LRS_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BIND_RCS_PRIORITY	LNMS\$SYSTEM_TABLE	4
RDM\$BUGCHECK_IGNORE_FLAGS	LNMS\$SYSTEM_TABLE	LP
SQL\$DATABASE	LNMS\$PROCESS_TABLE	*.RBF
RDM\$MAILBOX_CHANNEL	LNMS\$SYSTEM_TABLE	MBA80:
RDM\$MONITOR	LNMS\$SYSTEM_TABLE	SYS\$SYSROOT:[SYSEXE]

```
-----
```

```
Exit Full Help Menu >next_page <prev_page Set_rate Write !
```

The tabular display screens often contain multiple pages of information, as indicated by the “Page:” segment of the header region. To view the subsequent pages of the tabular display, press the right angle bracket (>) key, as indicated by the “>next_page” on-screen menu option in the on-screen menu. Then, to move back to previous screens, press the left angle bracket (<) key, as indicated by the “<prev_page” on-screen menu option.

DASHBOARD
 DISPLAY FORMAT

The dashboard display format is available for use exclusively by the “Database Dashboard” screens. For more information on this class of screens, please refer to Volume 4, Chapter 21.

The following is an example of the dashboard display format of the screen body region:

```
Node: MYNODE (1/1/1)      Oracle Rdb X7.0-00 Perf. Monitor  7-FEB-1998 06:56:26.54
Rate: 1.00 Second        IO Dashboard          Elapsed: 00:00:41.43
Page: 1 of 1            DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

Database..... Attribute.Name....	Current... Value.....	Previous.. Value.....	Lowest.... Value.....	Highest... Value.....	Original.. Value.....	Chng Cnt.
Buffer Count	20	20	20	20	20	0
APF Enabled	1	1	1	1	1	0
APF Depth	5	5	5	5	5	0
DAPF Enabled	1	1	1	1	1	0
DAPF Depth Count	4	4	4	4	4	0
DAPF Start Count	4	4	4	4	4	0
ABW Enabled	1	1	1	1	1	0
ABW Clean BufCount	5	5	5	5	5	0
ABW Batch Max	4	4	4	4	4	0

```
-----
Config Exit Help Menu Options Set_rate Write !
```

The layout of the dashboard display format is explained in the following list:

- Current Value** This column contains the current value for each database attribute. On global database screens, this is the default value being used by all processes. On per-process screens, this is the actual value being used by that particular process.
- Previous Value** This column contains the previous value for each database attribute. When a database attribute is dynamically updated, this column will contain the value before it was modified.
- Lowest Value** This column contains the lowest value for each database attribute.
- Highest Value** This column contains the highest value for each database attribute.
- Original Value** This column contains the original value for each database attribute. This column remains unchanged regardless of the changes made to the database attribute.
- Chng Cnt** This column contains the count of the number of changes to each database attribute. On global database screens, this count is the number of times the particular database

attribute was updated by the user. On per-process screens, this count is the number of times the actual process has updated its dynamic information.

The dynamic updates that users make to the database are reflected only in the "Current Value" column. The remaining columns reflect changes made by the current user only.

Screen Migration

Here is a list of the universal methods available to navigate through the various screens. Note that not all of these methods are available on each screen.

<u>Keystroke</u>	<u>Description</u>
⇒	Advance to next screen within current menu.
⇐	Advance to previous screen within current menu.
+	Advance to next “n” screens within current menu.
-	Advance to previous “n” screens within current menu.
>	Advance to next “page” of screen; does not wrap around at last page.
<	Advance to previous “page” of screen; does not wrap around at first page.
⇓	Advance to next “page” of screen; does wrap around at last page.
⇑	Advance to previous “page” of screen; does wrap around at first page.
#	Display the current menu.
[space]	Advance to next numeric screen within the current menu that exhibits activity. If there are no numeric screens within the current menu, or no screens within the current menu exhibit any activity, then simply advance to next screen within current menu, as if the ⇒ keystroke was used. This is an excellent method for finding any screen that contains database statistic activity.

GOTO SCREEN
 “BY-NAME”

This option allows you to enter the case-insensitive name of the screen you wish to go to, or any portion of the screen name. Note that certain screen names are not unique unless you have entered a sizable portion of the screen name. If you enter a response that is not unique, the first matching screen will be selected. Note that the selections are not necessarily in alphabetical order. If you enter a response that does not match any available screen, the RMU Show Statistic utility displays an error and you will be left on the current screen.

This screen migration option is available in the “Help” menu or the Tools menu.

GOTO SCREEN [→]

This option displays a multi-page menu of all available screen names, in alphabetical order. The selectable names of the screens depend on the screens available to the RMU Show Statistic utility. For example, the global buffer screen names are not selectable unless they are actually available.

This screen migration option is available in the “Help” menu or the Tools menu.

RMU Show Statistic Screen Overview

The following table provides an alphabetical list of the available RMU Show Statistic utility screens and a brief description of each. Please refer to “Volume 2” through “Volume 4” of this Handbook for more information on individual screens.

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
2PC Statistics	Provides information about distributed transaction performance.	Journaling Information
ABS Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
ACTIVE Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
ACTIVE Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
Active Stall Counts	Shows active stall activity broken down by stall types.	Process Information
Active User Chart	Shows the number of active users attached to the database over a period of time.	Process Information
Active User Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Active User Stall Messages	Shows all processes attached to the database on the current node and, if a process stalled, shows the reason why the stall occurred.	Process Information
AIJ Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
AIJ Backup Activity	Shows one line of information for each active AIJ Backup Server (ABS) or manual RMU Backup After_Journal utility on the node.	Process Information
AIJ Dashboard	Shows the actual database parameter and attribute settings used by the	Database

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	processes attached to the database.	Dashboard
AIJ Journal Growth Trend	Graphically portrays the size of the current after-image journal over a measured period of time.	Journaling Information
AIJ Journal Information	Shows information about all of a database's after-image journals on the current node.	Journaling Information
AIJ Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
AIJ Statistics	Shows a summary of after-image journaling activity.	Journaling Information
AIJDB Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
AIJFB Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
ALS Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
ALS Statistics	Shows AIJ log server activity.	Journaling Information
Area Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Area Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
Asynchronous IO Statistics	Shows statistics on asynchronous reads and writes to the database files.	IO Statistics Information
Audit Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Blocking ASTs	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
Buffer Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Buffer Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Buffer/Page Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
Cache Full	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Cache Searches	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Cache Unmark	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Checkpoint Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Checkpoint Information	Shows process checkpoint information.	Process Information
Checkpoint Statistics	Shows transaction and checkpoint activity.	Journaling Information
CLIENT Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
CLIENT Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
CLTSEQ Object ¹²	Shows statistics for a specific client sequence object collection category.	Objects (one stat type)
Collision	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
CPT Object	Shows statistics for a specific database rootfile object collection cate-	Objects (one

¹² Available in Oracle Rdb 8.0.

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	gory.	stat type)
CPU Utilization	Shows the current CPU utilization of each database process.	Process Information
Custom Statistics	Shows a customized screen of any of the base statistics information (for example, non-by-area and non-per-process statistics).	Main menu
DBKEY Information	Shows the last retrieved DBKEY for data, snapshot, SPAM, AIP, and ABM pages.	Process Information
DBR Activity	Shows one line of information for each database recovery process active on the node.	Process Information
DBR Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Defined Logicals	Shows a list of all the logical names currently accessible to the RMU Show Statistic utility.	Process Information
Device Information	Shows an online view of the storage area device information.	IO Statistics (by file)
Device IO Overview	Shows a composite view of I/O activity for all affected disk devices.	IO Statistics (by file)
Fast Commit Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Fast Incr Backup Statistics	Shows fast incremental backup activity.	Journaling Information
File IO Overview	Shows a composite view of I/O activity for all storage areas.	IO Statistics (by file)
File Lock Overview	Provides locking information for all storage areas.	Lock Statistics (by file)
FILID Locks	Shows lock statistics for one particu-	Locking (one

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	lar lock type.	lock type)
FILID Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
Found in Cache	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Found in Workset	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Found Too Big	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Freeze Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
GB Frequency Information	Displays the effectiveness of the global buffer pool for sharing of pages.	Global Buffer Information
GB Hot Page Information	Displays a list of the most heavily shared pages in the global buffer pool.	Global Buffer Information
GB Utilization	Shows the utilization of each page in a global buffer.	Global Buffer Information
General Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Group Commit Statistics	Provides information about AII group commit processing.	Journaling Information
Hash Index Statistics	Shows hashed index update and retrieval activity.	Index Information
Hash Misses	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Hot Row Information	Displays a list of the most frequently accessed rows for a specific row cache.	Row Cache Information

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
Hot Standby Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Hot Standby Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Hot Standby Statistics	Provides information about the performance and status of the hot standby feature.	Hot Standby Information
Index Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Index Statistics	Shows Oracle DBMS index information.	Index Menu
Index Statistics (Insertion)	Shows sorted index insertion and modification activity.	Index Information
Index Statistics (Removal)	Shows sorted index deletion activity.	Index Information
Index Statistics (Retrieval)	Shows sorted index retrieval activity.	Index Information
Insert Cache	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
IO Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
IO Stall Time	Shows a summary of all I/O stall activity.	IO Statistics Information
IO Statistics (by file)	I/O statistics for each database file.	IO Statistics (by file)
Journal Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Journaling Information	Shows dynamic information that automatically changes to reflect data-	Database Parameter Infor-

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	base parameter modifications.	mation
KROOT Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
Latch Requests	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Latch Retried	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Lock Deadlock History	Identifies the object that causes a deadlock event.	Process Information
Lock Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Lock Statistics (by file)	Provides information about page locks that are specific to storage areas and snapshot files.	Lock Statistics (by file)
Lock Timeout History	Identifies the object that causes a timeout event.	Process Information
Locking Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Locking Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Locks Demoted	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Locks Promoted	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Locks Released	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Locks Requested	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Logical Area Locks	Shows lock statistics for one particu-	Locking (one

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	lar lock type.	lock type)
Logical Area Overview	Shows database activity for all logical areas.	Logical Area Information
Logical Area Specific Displays	Shows database activity on a per-logical area basis.	Logical Area Information
MEMBIT Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
Monitor Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Monitor Log	Allows you to view the monitor log on line.	Process Information
Name translation	Shows statistics on database dashboard updates and logical name translation.	General Information
Nowait Transaction	Shows lock statistics for one particular lock type.	Locking (one lock type)
Objects Fetch Excl	Shows statistics for a specific database object.	Objects (one stat field)
Objects Fetch Shrd	Shows statistics for a specific database object.	Objects (one stat field)
Objects Modified	Shows statistics for a specific database object.	Objects (one stat field)
Objects Refreshed	Shows statistics for a specific database object.	Objects (one stat field)
Objects Released	Shows statistics for a specific database object.	Objects (one stat field)
Objects Written	Shows statistics for a specific database object.	Objects (one stat field)
OpenVMS SYSGEN Parameters	Shows dynamic information that automatically changes to reflect operating system parameter modifica-	Database Parameter Information

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	tions.	
Per-Process IO Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Per-process Journal Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Per-Process Row Cache Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
PIO Statistics--Data Fetches	Shows statistics on how data page requests are handled.	IO Statistics Information
PIO Statistics--Data Writes	Shows a summary of data file writes and buffer unmarking activity.	IO Statistics Information
PIO Statistics--File Access	Shows information about file access.	IO Statistics Information
PIO Statistics--SPAM Access	Shows statistics on what caused SPAM page requests.	IO Statistics Information
PIO Statistics--SPAM Fetches	Shows statistics on how SPAM page requests are handled.	IO Statistics Information
PIO Statistics--SPAM Writes	Shows SPAM write I/O information.	IO Statistics Information
Process Accounting	Shows continuously updated OpenVMS accounting information about local processes in a cluster environment.	Process Information
Process Analysis	Displays runtime process performance analysis information.	Online Analysis & Info.
Process IO Overview	Displays process-specific I/O summary information for all activated processes	Process Information
Process Journal IO Overview	Displays process-specific RUJ and AIJ journal I/O summary informa-	Process Information

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	tion for all activated processes	
Process Lock Overview	Displays process-specific locking summary information for all activated processes	Process Information
Process Object Overview	Displays process-specific rootfile object summary information for all activated processes	Process Information
Process Record Overview	Displays process-specific record summary information for all activated processes	Process Information
Process Snapshot Overview	Displays process-specific snapshot summary information for all activated processes	Process Information
Prom Deadlocks	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Prom Timeouts	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Proms Not Queued	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Proms Stalled	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Quiet Point Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
RCACHE Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat field)
RCS Statistics	Provides information about the runtime operation of the row cache server (RCS) process.	Row Cache Information
Record Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Record Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
Record Statistics	Shows a summary of data row activity.	Main Menu
Recovery Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Recovery Information	Provides information about the recovery performance of the hot standby feature.	Hot Standby Information
Recovery Statistics	Identifies various recovery phases and shows information on how long each phase took to complete.	Journaling Information
Row Cache (One Cache)	Provides summary information for a specific row cache.	Row Cache (One Cache)
Row Cache Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Row Cache Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
Row Cache Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Row Cache Queue Length	Provides information to determine the relative CPU performance impact of row caching.	Row Cache Information
Row Cache Status	Provides overall status for a specific row cache.	Row Cache Information
Row Cache Utilization	Provides utilization information for each row in a specific row cache.	Row Cache Information
Row Length Distribution	Shows the distribution of the various row lengths in a specific row cache.	Row Cache Information
Row Too Big	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Rqst Deadlocks	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
Rqst Timeouts	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Rqsts Not Queued	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Rqsts Stalled	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
RTUPB Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
RTUPB Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
RUI Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
RUI Dashboard	Shows the actual database parameter and attribute settings used by the processes attached to the database.	Database Dashboard
RUI Statistics	Provides summary information for all active update transactions.	Journaling Information
SEQBLK Lock	Shows lock statistics for one particular lock type.	Locking (one lock type)
SEQBLK Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
Skipped Dirty Slot	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Slipped Inuse Slot	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Snapshot Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
Snapshot Statistics	Shows snapshot activity for update and read-only transactions.	Main Menu
Stall Messages	Shows a summary of database users'	Process Infor-

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	stall activity.	mation
Stall Statistics	Shows a summary of stalls broken down by stall types.	Process Information
Stall Time x100	Shows lock statistics for all lock types of a particular statistical field.	Locking (one stat field)
Statistics Event Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Storage Area Information	Shows dynamic information that automatically changes to reflect database parameter modifications.	Database Parameter Information
Summary Cache Statistics	Shows cumulative information for all row caches in the database.	Main Menu
Summary Cache Unmark Statistics	Shows cumulative row cache unmark statistics that describe how rows in the row caches are written back to disk.	Main Menu
Summary IO Statistics	Shows a summary of database I/O activity.	Main Menu
Summary Locking Statistics	Shows a summary of the locking activity.	Main Menu
Summary Object Statistics	Shows cumulative information for all database root file objects.	Main Menu
Summary Tx Statistics	Shows a summary of database transaction activity.	Main Menu
Synchronization Mode Statistics	Provides a breakdown of each type of synchronization mode.	Hot Standby Information
Total Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
Transaction Analysis	Displays runtime performance analysis information.	Online Analysis & Info.
Transaction Duration	Shows overall realtime transaction	Main Menu

<u>Screen Name</u>	<u>Description</u>	<u>Menu</u>
	processing performance.	
Transaction Recovery Duration Estimate	Shows estimates of various transaction recovery durations.	Process Information
TSNBLK Locks	Shows lock statistics for one particular lock type.	Locking (one lock type)
TSNBLK Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
UTILITY Object	Shows statistics for a specific database rootfile object collection category.	Objects (one stat type)
Verb Statistics (1)	Shows Oracle DBMS verb information.	Main Menu
Verb Statistics (2)	Shows more Oracle DBMS verb information.	Main Menu
VLM Requests	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)
Window Turns	Shows row cache statistics for all row caches of a particular statistical field.	Row Cache (One Field)

CONFIG SCREENS

Most screens can be configured to provide more detailed or customized information. After pressing the “C” on-screen menu option (for “Config”), a menu of configuration options will be presented. Some screens provide one or two configuration options, while other screens provide many configuration choices.

Some screens, such as the columnar display format screens, cannot be configured.

Descriptions of individual screen configuration options are available for each screen in Volume 2 through Volume 4 of the Handbook.

ZOOM SCREENS

A zoom screen is a sub-window that displays detailed information about a specific screen item, typically a lock, process, after-image journal, page dump, row cache, or storage area. Note that a zoom screen is a static snapshot of information. Unlike normal statistics screens, the information does not change.

When a screen has zoom screen capability on a primary screen object, such as a process identifier or storage area name, the “Zoom” on-screen menu option will be displayed. Some screens contain secondary zoom screens on objects such as lock identifiers or DBKEYS; these screens display the “LockID” or “PageInfo” on-screen menu options.

Zoom screens will automatically close themselves based on the user-prompt timeout interval. This interval defaults to one minute. You can modify the user-prompt timeout interval using the “Set user-prompt timeout” option in the Tools menu.

It is often necessary to quickly review detailed information about a storage area or active database process. Typically, this need occurs while you are reviewing one of the by-area or per-process screens and some event occurs that warrants further investigation.

For example, the following “Checkpoint Information” screen is interesting because one of the processes, 2020EC26, has not checkpointed:

```
Node: MYNODE (1/1/1)      Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 0.10 Seconds      Checkpoint Information          Elapsed: 00:28:45.14
Page: 1 of 1           DISK$: [DB_HOMEDIR.WORK.ALS]MF_PERSONNEL.RDB;12   Mode: Online
-----
Process.ID Ckpt.Vno:Ckpt.Vbn QuietVno
2020D41F:1s      32      106      0
2020EC26:1s      33      132      0
20208957:1      33      119      0
20207EF7:1      33      135      0
202084F9:1      33      135      0
-----
Config Exit Help Menu >next_page <prev_page Pause Set_rate Write Zoom !
```

Selecting the “Zoom” on-screen menu option provides information on why process 2020EC26 has not checkpointed. After pressing the Z key, a menu of the currently displayed process identifiers is presented; select the desired process identifier. The RMU Show Statistic utility then displays detailed information about the specified process.

A zoom screen similar to the following is displayed:

```
Node: MYNODE (1/1/1)   Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 0.10 Seconds    Checkpoint Information           Elapsed: 00:30:26.38
Page: 1 of 1         DISK$:[DB_HOMEDIR.WORK.ALS]MF_PERSONNEL.RDB;12   Mode: Online
```

```
-----
Process.ID Ckpt.Vno:Ckpt.Vbn QuietVno
2020D41F:1s      34      89      0
2020EC26:1s      34      89      0
20208957:1      34     164     0
20207EF7:1      34     169     0
202084F9:1      34     163     0
```

```
+--Process Information: 2020EC26-----+
|
| Active user with process ID 2020EC26 (database server)
| User name is RDBVMS
| Process name is RDM_LCS701_1
| Image name is DSA1:[SYS1.SYSCOMMON.][SYSEXE]RDMLCS701.EXE;206
| No transaction in progress
| Monitor ID is 1
| Internal Stream ID is 1
| Internal Transaction ID is 655
|
+-----+
```

```
-----
Exit Help Menu >Next_page <Prev_page Pause Set_rate Write Zoom !
```

As you can see, process 2020EC26 is a database server process and does not have a transaction in progress.

ZOOM SCREEN
NAVIGATION

The zoom screen displays often contain more information than can be displayed on the terminal screen. Therefore, you can scroll through the zoom screen display either one line at a time or one screen at a time.

You can move the zoom screen up or down one line at a time using the up-arrow (↑) or down-arrow (↓) keys. You can also move the lock information up or down one page at a time using the Previous Screen or left-arrow (←) key, or the Next Screen or right-arrow (→) key. These keys are activated when the indicators “←MORE” and/or “MORE→” dynamically appear on the lock information display itself. The on-screen menu at the bottom of the screen also indicates whether these keys are activated. If the navigational keys are not activated, nothing happens if you attempt to use them.

Here is a list of the universal methods available to navigate through the various pages of a zoom screen:

<u>Keystroke</u>	<u>Description</u>
-------------------------	---------------------------

<u>Keystroke</u>	<u>Description</u>
Up-arrow (↑) or Down-arrow (↓)	Displays the previous or next line of the zoom screen
Left-arrow (←) or Right-arrow (→)	Displays the previous or next page of the zoom screen.
W	Writes the current screen to a text file named RMU.SCR.
Ctrl/W	Repaints the zoom screen.
Ctrl/Z	Cancel the zoom screen.

Note that the zoom screen display size is based on the number of lines configured for the terminal. For instance, a terminal with 48 lines has a larger zoom screen display than a terminal with 24 lines.

PROCESS
INFORMATION

This zoom screen displays information about a particular process.

The following is an example of a process zoom screen:

```
Node: MYNODE (1/1/1)      Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 0.10 Seconds      Checkpoint Information           Elapsed: 00:30:26.38
Page: 1 of 1           DISK$:[DB_HOMEDIR.WORK.ALS]MF_PERSONNEL.RDB;12      Mode: Online
```

```
-----
Process.ID Ckpt.Vno:Ckpt.Vbn QuietVno
2020D41F:1s      34      89      0
2020EC26:1s          0
20208957:1      34      164     0
20207EF7:1      34      169     0
202084F9:1      34      163     0
```

```
+--Process Information: 2020EC26-----+
|
| Active user with process ID 2020EC26 (database server)
| User name is RDBVMS
| Process name is RDM_LCS701_1
| Image name is DSA1:[SYS1.SYSCOMMON.][SYSEXE]RDMLCS701.EXE;206
| No transaction in progress
| Monitor ID is 1
| Internal Stream ID is 1
| Internal Transaction ID is 655
|
+-----+
```

```
-----
Exit Help Menu >Next_page <Prev_page Pause Set_rate Write Zoom !
```

The contents of the zoom screen should be self-explanatory, and are very similar to the RMU Dump Header utility output.

AIJ JOURNAL
INFORMATION

This zoom screen displays information about a particular after-image journal.

The following is an example of an after-image journal zoom screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 12:57:43.18
Rate: 1.00 Second      AIJ Journal Information           Elapsed: 04:45:42.00
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
```

```
-----
Journaling: enabled   Shutdown:    60  Notify: enabled   State: A ccessible
ALS: Manual          ABS: enabled  ACE: disabled  FC: enabled   CTJ: disabled
ARB.Count: 1001     ARB.Avail: 1001
After-Image.Journal.Name..... SeqNum AIJsize CurrEOF Status. State.....
RICK1                Unused    1024   Empty Latent A ccessible
RICK2                Unused    1024   Empty Latent A ccessible
RICK3                83       1026   181 Current A ccessible
RICK4                Unused    1024   Empty Latent A ccessible
RICK5                Unused    1024   Empty Latent A ccessible
Available AIJ slot 1
Available AIJ slot 2
Available AIJ slot 3
Available AIJ slot 4
Available AIJ slot 5
Available AIJ slot 6
```

```
+-----+
| AIJ Journal "RICK1"
| Filename is "KODA_TEST:[DB_HOMEDIR.TMP]RICK1.AIJ;1"
| Default AIJ file name is "KODA_TEST:[DB_HOMEDIR.TMP]RICK1.AIJ"
| Backup sequence number is -1
| Allocation is specified as 1024 blocks
| Extension is defaulted to 1024 blocks
| Backup file name is defaulted to "KODA_TEST:[DB_HOMEDIR.TMP]BACKUP.AIJ"
| Backup file name edits are disabled
+-----+
```

```
-----
Bell Exit Help Menu >next_page <prev_page Refresh Set_rate Write Zoom !
```

The contents of the zoom screen should be self-explanatory, and are very similar to the RMU Dump Header utility output.

STORAGE AREA
INFORMATION

This zoom screen displays information about a particular database storage area.

The following is an example of storage area zoom screen:

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 14:54:51.30
Rate: 1.00 Second File IO Overview (Unsorted total I/O) Elapsed: 06:42:50.12
Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

File/Storage.Area.Name.....	Sync.Reads	SyncWrites	AsyncReads	AsyncWrites	PgDis
Database Root	79	570	0	68025	0
AIJ (After-Image Journal)	3204	57969	0	0	0
RUJ (Recovery-Unit Journal)	3	14	0	27301	0
ACE (AIJ Cache Electronic)	0	0	0	0	0
All data/snap files	1436288	98357	744	89472	222k
data MF_PERS_DEFAULT	98550	3581	6	25059	26417
data DEPARTMENTS	2	1	0	0	0
data EMPIDS_LOW	30	34	136	0	0
data EMPIDS_MID	47	48	136	0	0
data EMPIDS_OVER	1268659	12572	188	6441	196k
data EMP_INFO	2	5	270	0	0
data JOBS	0	0	0	0	0
data MF_PERS_SEGSTR	83	1	0	0	13
data SALARY_HISTORY	7	12	8	0	0
snap MF_PERS_DEFAULT	26514	27801	0	0	0
snap DEPARTMENTS	1	1	0	0	0
snap EMPIDS_LOW	11	11	0	0	0
snap EMPIDS_MID	18	18	0	0	0
snap EMPIDS_OVER	43488	54248	0	0	0
snap EMP_INFO	4	4	0	0	0
snap JOBS	0	0	0	0	0
snap MF_PERS_SEGSTR	1	1	0	0	0
snap SALARY_HISTORY	19	19	0	0	0

```

+-----+
| Storage area "EMPIDS_OVER"
| Area ID number is 5
| Filename is "DISK$:[DB_HOMEDIR.WORK.STATS]EMPIDS_OVER.RDA;1"
| Access mode is read/write
| Page format is mixed
| Page size is 2 blocks
| - Current physical page count is 2592
| Record level locking is enabled
| Row caching is enabled
| No row cache is defined for this area
| Extends are enabled
| - Extend area by 20%, minimum of 99 pages, maximum of 9999 pages
| - Area has been extended 10 times
| Volume set spreading is enabled
| Snapshot area ID number is 34
| SPAMs are enabled
| - Interval is 216 data pages
| - Thresholds are 70%, 85%, and 95%
| - Current SPAM page count is 12
| Area last backed up at 26-JAN-1998 08:10:27.42
| Area has never been incrementally restored
+-----+

```

Config Exit Filter Help Menu >next_page <prev_page Options Reset Set_rate Write

The contents of the zoom screen should be self-explanatory, and are very similar to the RMU Dump Header utility output.

ROW CACHE
 INFORMATION

This zoom screen displays information about a particular row cache or the contents of a row cache.

The following is an example of a row cache zoom screen:

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 15:31:28.26
 Rate: 1.00 Second Hot Row Information Elapsed: 00:08:36.18
 Page: 1 of 40 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

```
-----
```

For Cache: EMPIDS_OVER (unsorted)

Area:Page:Ln	#Users	State	Length	SlotNo	Area:Page:Ln	#Users	State	Length	SlotNo
Reserved	1		0	0	Empty	0		0	38
Reserved	1		0	1	Empty	0		0	39
Reserved	1		0	2	Empty	0		0	40
Reserved	1		0	3	Empty	0		0	41
Reserved	1		0	4	Empty	0		0	42
Reserved	1		0	5	Empty	0		0	43
Reserved	1		0	6	Empty	0		0	44
Reserved	1		0	7	Empty	0		0	45
Reserved	1		0	8	Empty	0		0	46
Reserved	1		0	9	Empty	0		0	47
Reserved	1		0	10	Empty	0		0	48
Reserved	1		0	11	Empty	0		0	49
Reserved	1		0	12	Empty	0		0	50
Reserved	1		0	13	Empty	0		0	51
Reserved	1		0	14	Empty	0		0	52
Reserved	1		0	15	Empty	0		0	53
Reserved	1		0	16	Empty	0		0	54
Reserved	1		0	17	Empty	0		0	55
Reserved	1		0	18	Empty	0		0	56
Reserved	1		0	19	Empty	0		0	57
54:232:0	1		5	20	Empty	0		0	58
Empty	0		0	21	Empty	0		0	59
Empty	0		0	22	Empty	0		0	60
Empty	0		0	23	Empty	0		0	61
Empty	0		0	24	Empty	0		0	62
Empty	0		0	25	Empty	0		0	63
Empty	0		0	26	Empty	0		0	64
Empty	0		0	27	Empty	0		0	65
Empty	0		0	28	Empty	0		0	66
Empty	0		0	29	Empty	0		0	67
Empty	0		0	30	Empty	0		0	68
Empty	0		0	31	Empty	0		0	69
Empty	0		0	32	Empty	0		0	70
Empty	0		0	33	Empty	0		0	71
Empty	0		0	34	Empty	0		0	72
Empty	0		0	35	Empty	0		0	73
Empty	0		0	36	Empty	0		0	74
Empty	0		0	37	Empty	0		0	75

```
+-----+
| 54:232:0 LEN=5 TSN=0:0 SnapPage=-1 |
| 2001 8568 line 0 (54:232:0) SYSTEM record |
| 00 0001 856A 1 byte in 0 sets/dynamic items |
+-----+
```

 Config Exit Help Menu >next_page <prev_page Options Set_rate Write Zoom !

This zoom screen displays the following information:

Information	Status
DBKEY	This field identifies the selected DBKEY
LEN	This field identifies the actual row length
TSN	This field identifies the TSN of the previous modifying transaction
SnapPage	This field identifies the page number of the corresponding snapshot record.
Contents	This field displays the binary contents of the selected row

LOCK
INFORMATION

This zoom screen is activated using the “LockID” on-screen menu option, and displays information about a particular stalled lock.

The RMU Show Statistic utility provides integrated lock information on the “Stall Messages”, “Active User Stall Messages”, and “DBR Activity” screens.

For example, consider the following “Active User Stall Messages” screen:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 1.00 Second Active User Stall Messages Elapsed: 03:06:07.20
Page: 1 of 1 KODD$: [DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1 Mode: Online
-----
Process.ID Since..... Stall.reason..... Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX) 1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX) 1200A2E4
2B600555:1 writing pages back to database
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

Typing “L” displays the menu of available lock identifiers. For example:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 1.00 Second Active User Stall Messages Elapsed: 03:06:07.20
Page: 1 of 1 KODD$: [DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1 Mode: Online
-----
Process.ID Since..... Stall.reason..... Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX) A. 1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX) B. 1200A2E4
2B600555:1 writing pages back to database
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

Note the displayed lock identifiers have been prefixed with the standard menu-selection letter choice. The desired lock identifier can be selected by either typing the desired letter, or moving the highlighted cursor using the up-arrow (↑) or down-arrow (↓) keys and then pressing the Return key to select the highlighted lock.

If there are no lock identifiers on the display, typing “L” has no effect.

When the desired lock identifier is selected, the “Lock Information” zoom screen appears. For example:

```

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 1.00 Second Active User Stall Messages Elapsed: 03:06:07.20
Page: 1 of 1 KODD$:[DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1 Mode: Online
-----
Process.ID Since..... Stall.reason..... Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX) 1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX) 1200A2E4
2B600555:1 writing pages back to database
+-- Lock Information: 1500624B -----+
|
| Resource: record 321:2
| State... ProcessID Process.Name... Lock.ID. Rq Gr Queue
|
| Blocker: 2B600555 RICK3..... 0F005AB6 EX EX Grant
| Waiting: 2B600556 RICK4..... 1200A2E4 EX NL Cnvrt
| Waiting: 2B600554 RICK2..... 1500624B EX NL Cnvrt
|
+-----+
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
  
```

The “Lock Information” display is not dynamic. The lock information is captured at the time the lock identifier is selected, and the information does not dynamically change. The “Lock Information” zoom screen is a snapshot in time of the selected lock identifier.

The “Lock Information” zoom screen can also be saved using the “Write” on-screen menu option.

The “Lock Information” zoom screen provides the following information:

<u>Information</u>	<u>Description</u>
State	This field indicates whether the respective lock is an owner, a blocker, or is waiting for the resource.
ProcessID	This field indicates the identifier of the process that owns the respective lock.
Process.Name	This field indicates the name of the process that owns the

<u>Information</u>	<u>Description</u>
	respective lock.
Lock.ID	This field indicates the identifier of the lock on the indicated resource.
Rq	This field indicates the mode in which the respective lock was requested. Modes are NL, CR, CW, PR, PW, and EX.
Gr	This field indicates the mode in which the respective lock was granted. Modes are NL, CR, CW, PR, PW, and EX. The granted mode may differ from the requested mode if the lock is on the conversion queue or if the lock was demoted from a stronger to a weaker mode.
Queue	This field indicates the queue on which the respective lock resides. There are three queues:
	Grant the granted lock queue
	Cnvt the conversion lock queue
	Wait the waiting lock queue

Note that while the order of displayed locks is arbitrary, you can identify the order in which waiting locks will be granted. For instance, in the previous example, two processes are waiting for the same lock. Displaying the information for lock 1500624B simply indicates that both processes are waiting for the same resource.

Displaying the information for lock 1200A2E4 shows the following information:

```

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 1.00 Second Active User Stall Messages Elapsed: 03:06:07.20
Page: 1 of 1 KODD$:[DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1 Mode: Online
-----
Process.ID Since..... Stall.reason..... Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX) 1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX) >1200A2E4<
2B600555:1 writing pages back to database
+-- Lock Information: 1500624B -----+
|
| Resource: record 321:2
| State... ProcessID Process.Name... Lock.ID. Rq Gr Queue
|
| Blocker: 2B600555 RICK3..... 0F005AB6 EX EX Grant
| Blocker: 2B600554 RICK2..... 1500624B EX NL Cnvrt
| Waiting: 2B600556 RICK4..... 1200A2E4 EX NL Cnvrt
|
+-----+
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !

```

This display indicates that, although lock 1200A2E4 is on the conversion queue, it is blocking the other lock and will be granted before the other lock.

LOGICAL AREA
INFORMATION¹³

This zoom screen is activated using the “Zoom” on-screen menu option from the “Logical Area Overview” screen. It displays statistic information about a selected logical area, such as a table b-tree index, hash index or blob.

The “Logical Area Information” zoom screen information is not dynamic. The statistic information is captured at the time the logical area is selected, and the information does not dynamically change. The “Logical Area Information” zoom screen is a snapshot in time of the selected logical area.

The logical area name and its corresponding storage area name are identified on the zoom screen border.

¹³ Available in Oracle Rdb 7.0.2.0.

The following is an example of the “Logical Area information” zoom screen:

Node: ALPHA3 (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 20-MAR-1998 09:11:01.54
Rate: 1.00 Second Logical Area Overview (Tables) Elapsed: 01:21:52.46
Page: 1 of 1 KODH\$:[R_ANDERSON.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

```
-----
```

Logical.Area.Name...	record fetch	record store	record erase	discarded	CurTot
CANDIDATES.RDB\$SYSTE	0	0	0	0	
COLLEGES.EMP_INFO	0	0	0	0	
DEGREES.EMP_INFO	0	0	0	0	
DEPARTMENTS.DEPARTME	0	0	0	0	
EMPLOYEES.EMPIDS_LOW	0	0	0	0	
EMPLOYEES.EMPIDS_MID	0	0	0	0	
EMPLOYEES.EMPIDS_OVE	8192	8192	0	2531	
JOBS.JOBS	0	0	0	0	
JOB_HISTORY.EMPIDS_L	0	0	0	0	
JOB_HISTORY.EMPIDS_M	0	0	0	0	
JOB_HISTORY.EMPIDS_O	0	0	0	0	
RESUMES.EMP_INFO	0	0	0	0	
SALARY_HISTORY.SALAR	0	0	0	0	
WORK_STATUS.EMP_INFO	0	0	0	0	

```
+-- Logical Area Information: EMPLOYEES in EMPIDS_OVER -----+
```

statistic.....	rate.per.second....	total.....	ave rage.....	
name.....	max.....	avg.....	count.....	per.trans....
record marked	0	0.0	0	0.0
record fetched	1	1.3	8192	0.9
fragmented	0	0.0	0	0.0
record stored	1	1.3	8192	0.9
fragmented	0	0.0	0	0.0
pages checked	1	1.7	10723	1.3
saved IO	1	1.0	6378	0.7
discarded	0	0.4	2531	0.3
record erased	0	0.0	0	0.0
fragmented	0	0.0	0	0.0

```
+-----+
```

Config Exit Help Menu >next_page <prev_page Options Pause Reset Set_rate Write

The “Logical Area Information” zoom screen can also be saved using the “Write” on-screen menu option.

The “Logical Area Information” zoom screen provides the following information:

<u>Information</u>	<u>Description</u>
statistic name	The name of a statistic field appropriate for the selected logical area.
rate per second (max)	This column indicates the maximum occurrence-per-second rate of the statistic since the RMU Show Statistic utility was started or the statistic collection information was reset; you can reset this information using the “Reset” on-screen

<u>Information</u>	<u>Description</u>
	menu option.
rate per second (avg)	This column indicates the average occurrence-per-second rate of the statistic since the database was opened or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
total count	This column indicates the total number of occurrences of the statistic since the database was opened or the statistic collection information was reset; you can reset this information using the “Reset” on-screen menu option.
average per trans	This column indicates the transaction average of the total-count column and the total number of completed transactions.



Note that there is no “rate per second (cur)” statistic information displayed. Because a zoom screen represents a snapshot in time of the statistic information for the selected logical area, there cannot be any current statistic information.

PAGE
 INFORMATION

This zoom screen is activated using the “PageInfo” on-screen menu option, and displays information about a particular database page.

For security reasons, the actual contents of each record image cannot be displayed. You must use the RMU Dump utility to view the actual record contents.

The “Page Information” zoom screen is currently available from the “Stall Messages”, “Active User Stall Messages”, “DBR Activity” and “DBKEY Information” screens.

The “Page Information” zoom screen is selected using the “PageInfo” on-screen menu option, by pressing the “**P**” key.







You will be prompted to select a process from the list of available processes with DBKEY information displayed on the screen. Only those processes displaying DBKEY information can be selected.

If the process you selected is accessing a range of pages, you will be prompted to select the desired page from a submenu provided.

If you are displaying page information from the “DBKEY Information” screen, you will be prompted to select one of the types of pages being accessed by that process.

It is also possible to display an arbitrary page using the Tools menu, obtained using the exclamation point (!). Simply select the “Display page information” option and you will be prompted for the desired storage area and page number.

The following caveats apply to the “Page Information” zoom screen:

-  For security reasons, the contents of individual lines on a data page cannot be displayed. The contents of area inventory pages cannot be displayed, either. Contact your DBA for other methods to display the contents of selected rows.
-  Because the page information can be quite lengthy, you are able to migrate through the various pages using the standard zoom screen migration keystrokes.
-  Of course, the page information “zoom” screen contents can be written to disk using the “Write” on-screen menu option.
-  The “PageInfo” on-screen menu option is not available during replay of a binary input file.
-  No locking of the selected database page actually occurs. Therefore, it may be possible, but unlikely, to display inconsistent page information.
-  The display of logical area DBKEYs is not possible if the NOLOGICAL_AREA qualifier was specified.

The following is an example of a live data page “Page Information” zoom screen:

```

+-----+
0001 00000005 0000 page 5, physical area 1 (data)
      8EE86A99 0006 checksum = 8EE86A99
009BA463 0DA1FC74 000A time stamp = 14-SEP-1997 06:51:12.75
      0000 006A 0012 106 free bytes, 0 locked
          0002 0016 2 lines
      01AE 0240 0018 line 0: offset 0240, 430 bytes
      01AE 0092 001C line 1: offset 0092, 430 bytes
          0000018C 0020 line 0: TSN 396
          000001BF 0024 line 1: TSN 447
          00000024 03EE snap page pointer 36
          000001BF 03F2 snap pointer TSN 447
          003B 03F6 logical area 59
      0000003B 03F8 page sequence number 59
          0000 03FC page TSN base 0
          0000 03FE MBZ '..'
+-----+
    
```

The following is an example of a snapshot data page "Page Information" zoom screen:

```

4001 00000001 0000 page 1, physical area 1 (snap)
      A46ACD6A 0006 checksum = A46ACD6A
009BA304 993A8B26 000A time stamp = 12-SEP-1997 13:02:33.60
      0000 0054 0012 84 free bytes, 0 locked
          0003 0016 3 lines
0000 0000 0018 line 0: empty
01AE 0238 001C line 1: offset 0238, 430 bytes
01AE 008A 0020 line 2: offset 008A, 430 bytes
      00000000 0024 line 0: TSN 0
      000085BD 0028 line 1: TSN 34237
      000085BF 002C line 2: TSN 34239
          0000 0030 line 0 -> live line: 0
          0000 0032 line 1 -> live line: 0
          0000 0034 line 2 -> live line: 0
000001AB 03E6 live page pointer 427
000085C4 03EA max TSN 34244
FFFFFFF 03EE snap page pointer -1
00000000 03F2 snap pointer TSN 0
      0000 03F6 MBZ '..'
00000000 03F8 page sequence number 0
      0000 03FC page TSN base 0
      0000 03FE MBZ '..'

```

The following is an example of an area inventory page (AIP) page information zoom screen:

```

0001 000001D0 0000 page 464, physical area 1 (AIP)
      D992664E 0006 checksum = D992664E
009646FF 8BFE44E0 000A time stamp = 1-DEC-1992 12:49:48.
      0000 0022 0012 34 free bytes, 0 locked
          000001D1 0016 next area inventory page 465
              4001 03F6 logical area 16385
00000000 03F8 page sequence number 0
      0000 03FC page TSN base 0

```


The following is an example of a SPAM page “Page Information” zoom screen; note that a SPAM page display is quite voluminous:

```

0001 00000001 0000 page 1, physical area 1 (SPAM)
          4681E156 0006 checksum = 4681E156
80000000 00000060 000A Fast incremental backup TSN = 0:96
          0000 0001 0012 1 free byte, 0 locked
FFFFFFFFFFFFFFFFCFFFFFFFFFFFFFFFFF 0016 pages 2-31: threshold 3
          page 32: threshold 0
          pages 33-65: threshold 3
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0026 pages 66-129: threshold 3
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0036 pages 130-193: threshold 3
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0046 pages 194-257: threshold 3
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF 0056 pages 258-321: threshold 3
0FFFFFFC33C3FFFFFFFFFFFFFFFFFFFFFFFF 0066 pages 322-362: threshold 3
          pages 363-364: threshold 0
          pages 365-366: threshold 3
          page 367: threshold 0
          page 368: threshold 3
          pages 369-370: threshold 0
          pages 371-383: threshold 3
          pages 384-385: threshold 0
33FFFFFFFF03FFFFFFC03FFFFFF3FF0FFFFF 0076 pages 386-395: threshold 3
          pages 396-397: threshold 0
          pages 398-402: threshold 3
          page 403: threshold 0
          pages 404-414: threshold 3
          pages 415-418: threshold 0
          pages 419-430: threshold 3
          pages 431-433: threshold 0
          pages 434-446: threshold 3
          page 447: threshold 0
          page 448: threshold 3
[ repeated lines deleted for brevity ]
0052 03EF pages 1067-1069, logical area 82
0052 03F1 pages 1070-1072, logical area 82
0052 03F3 pages 1073-1075, logical area 82
0052 03F5 pages 1076-1078, logical area 82
0052 03F7 pages 1079-1081, logical area 82
0052 03F9 pages 1082-1084, logical area 82
0052 03FB pages 1085-1087, logical area 82
0052 03FD pages 1088-1090, logical area 82
00 03FF MBZ free '.'

```

TIME PLOT
SCREEN ANALYSIS

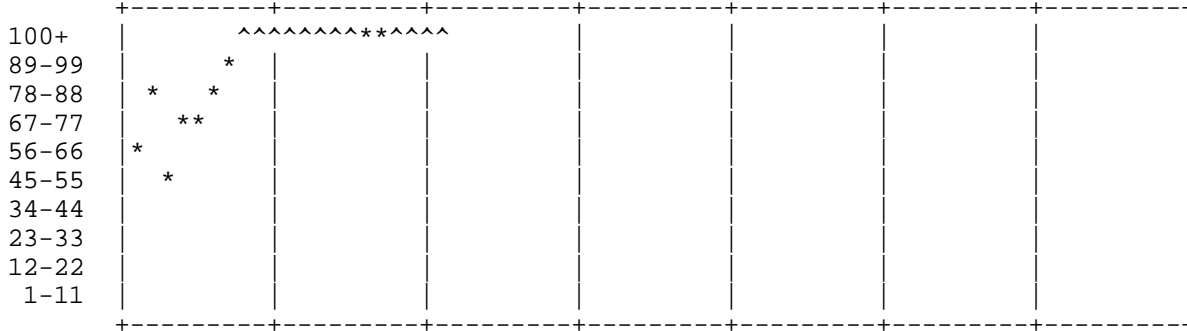
The Time Plot display format provides a detailed graph of event counts for a particular field on the screen. Each screen includes an on-screen menu. If the on-screen menu lists the “Time_plot” option, you can type “T” to request a time plot. After you type “T”, select a statistic field from the screen by typing the letter associated with the desired field, or by using the arrow keys to move the cursor to the desired field and then pressing the Return key.

The following display is an example of a time plot that shows verb success rates over time:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds Summary IO Statistics Elapsed: 00:07:35.33
Page: 1 of 1 SQL$DISK1:[USER]MF_PERSONNEL.RDB;1 Mode: Online
```

verb successes

Maximum rate = 147 Current rate = 144 Average rate = 78.0
Total count = 10216 Average per transaction = 8.9



Sample interval is 3.00 seconds

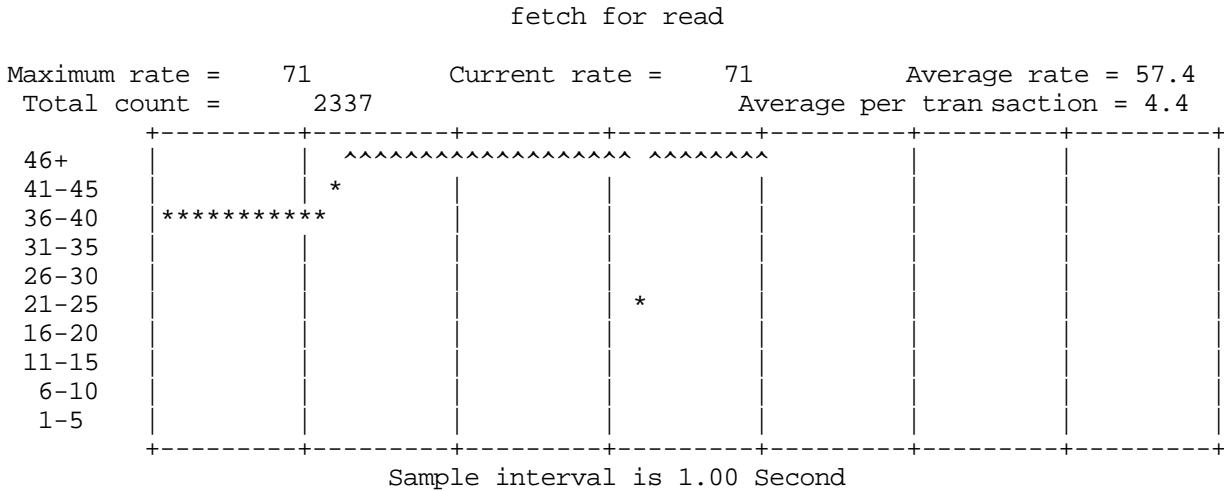
Exit Help Menu Normal Pause Reset Set_rate Unreset Write !

The circumflex characters (^) in the top row indicate that the number of events (verb successes in the previous example) is larger than the highest range value of the chart. To change the range of the display, select the "Reset" on-screen menu option

If you change the sample interval using the "Set_rate" on-screen menu option, the display is rescaled to the new interval.

For example, the following “PIO Statistics– Data Fetches” screen display uses a sample interval of one second:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 09:44:44.97
Rate: 1.00 Second      PIO Statistics--Data Fetches      Elapsed: 00:00:40.68
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
```



```
-----
Exit Help Menu Normal Pause Reset Set_rate Unreset Write !
```

The Tools submenu provides an option that allows you to enable or disable the Time_plot baseline. Enabling the baseline updates the graph continuously. You can enable the Time_plot baseline by performing the following steps:

1. Enter the exclamation point (!) to invoke the Tools menu.
2. Select the “SHOW STATISTIC options [→“ option from the Tools submenu.
3. Select “Enable Time_plot baseline” option from options submenu.

SCATTER PLOT
SCREEN ANALYSIS

The Scatter Plot display format, similar to the Time Plot display format, shows information for a selected statistic field on the current screen. However, the information in the scatter plot is displayed in a vertical histogram. The Scatter Plot display format allows you to determine the composition of the values that determine the average rate statistic.

If the on-screen menu lists the “X_plot” option, you can type “X” to request a scatter plot. After you type “X”, select a statistic field from the screen by typing the letter associated with the desired field (or by using the arrow keys to move the cursor to the desired field and then pressing the Return key).

After you have selected a field, the Scatter Plot display initially shows that the scale of the display is based on the running average rate for the selected field. This scale can be changed using the "Reset" on-screen menu option.

Each vertical line of the screen identifies a collection rate bucket based on the selected statistic rate. The complete set of buckets is known as the collection range and comprises the entire Scatter Plot screen. For example, the Scatter Plot screen may have a range from 0.0 to 1.5 seconds, with each bucket representing 0.1 (one-tenth) of a second.

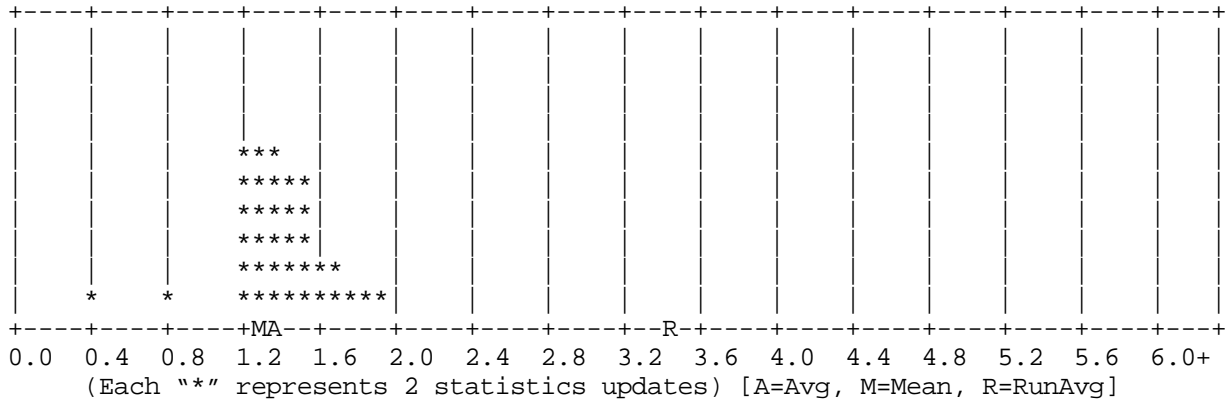
Based on the selected screen refresh rate, the normalized current rate for the selected statistic field is examined to determine into which bucket the rate would occur. This algorithm results in a current rate scattered distribution graph, which is extremely beneficial for performance analysis.

The following example shows a Scatter Plot display that is interesting because the current rate distribution is significantly more than the running average (indicated by the letter "R" on the bottom horizontal axis). This indicates that some event has recently occurred that resulted in sudden increased asynchronous data read activity. Also, notice the skewed distribution of the data.

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 26-JAN-1998 09:36:24.03
Rate: 1.00 Second      PIO Statistics--SPAM Access       Elapsed: 01:24:22.85
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
```

```
-----
Out-of-range count:      0.3--- = 0                     6.0+++ = 0
Current statistic range: low = 0.5                     high = 1.9
```

```
                        fetch for read
Scaled distribution of statistic rate (in seconds)          Elapsed: 00:01:15.48
```



```
-----
Config Exit Help Menu Normal Pause Reset Set_rate Write !
```

Three sliding indicators are displayed along the bottom horizontal axis:

<u>Letter</u>	<u>Description</u>
R	The letter R indicates the running average rate, which is computed over the total elapsed time as shown in the summary region. Note that the R running

<u>Letter</u>	<u>Description</u>
	average rate may not always appear, depending on the collection range. Also, resetting the elapsed time is useful for comparing the running average against the collected average.
A	The letter A indicates the collected average rate, which is computed on the collection time, displayed as the “elapsed” time in the body region of the screen.
M	The letter M indicates the collected median rate, which is also computed on the collection time, displayed as the “elapsed” time in the body region of the screen.

The number of collection buckets is based on the display width of the terminal. By default, each bucket comprises five columns of the display. The number of columns can be configured by the user. For example, setting the terminal width to 132 columns allows you to display more information than with an 80-column terminal width. Conversely, decreasing the number of columns per bucket achieves the same effect on an 80-column terminal width, but with a corresponding loss of precision.

Notice that the collected median can be significantly higher than the collected average. This is indicative of skewed data, typically caused by periodic blips in the runtime performance. Identifying and eliminating these blips produces a smoother-running system.

Notice also that these occasional blips tend to get lost as noise in the running average. However, removing these blips ultimately improves the overall running average of the system.

COLUMNS PER
BUCKET

Decreasing the number of columns per bucket allows you to collect more rates. Conversely, increasing the number of columns per bucket allows you to collect fewer rates, but with more accurate rate reporting (that is, better scaling) per bucket. The minimum number of columns is 2 and the maximum is 10.

Setting specific collection range values allows you to determine the actual collection range, which is normally based on the rates already collected.

The Scatter Plot screen information is recorded in the binary output file created using the OUTPUT qualifier, and can be replayed using the INPUT qualifier. Note that changes in collection rates during the recording phase are not identified during replay and may cause interesting Scatter Plot displays.

Designing a Custom Screen

The RMU Show Statistic utility allows you to create a customized statistics screen. Any of the base numeric statistics information (for example, non-by-area and non-per-process) can be customized.

The size of your “Custom Statistics” screen determines the number of statistic fields you can add. The header and on-screen menu take up eight lines of the Custom Statistics screen; you can add a statistic field on each of the remaining lines, up to a maximum of 36 different fields. Each statistic field is placed on a separate line of the screen.

The Custom Statistics screen is an ordinary screen; information can be displayed graphically as a histogram or in numeric tabular format. Also, the “Time_plot” and “X_plot” on-screen menu options are available for any of the selected statistic fields.

CUSTOM SCREEN
LOCATION

To select the “Custom Statistics” screen, select the “Menu” on-screen menu option and then move the cursor with the down arrow key to the “Custom Statistics” option and press Return.

Initially, the Custom Statistics screen contains two statistics: database binds and transactions. These statistics can be moved, removed, or replaced.

CUSTOM SCREEN
METHODOLOGY

There are two different methods available to populate the “Custom Statistics” screen. You can “yank” statistic fields from an existing statistics screen and implicitly put the selected statistic fields on the “Custom Statistics” screen. You can also manually create a statistic field. However, Oracle Corporation recommends using the yank-and-put method, which is considerably easier.

To yank and put a statistic field, use the Yank menu option on the statistics screen containing the statistic field you want to select. The “Yank” on-screen menu option is not available in the “Custom Statistics” screen.

When you select “Y”, a menu of the selectable statistic fields is shown; this menu is similar to the “Help on fields” menu. Select the letter of the statistic field you want to yank and that field will automatically appear on the “Custom Statistics” screen. Note that once you select a particular statistic field, you cannot select it again unless you first delete it from the “Custom Statistics” screen. This means that the field selection menu changes every time you select a field.

The selected field is positioned on the first available row of the “Custom Statistics” screen. You cannot specify the row of the “Custom Statistics” screen on which the field will reside using the yank-and-put method. However, once the field has been put on the “Custom Statistics” screen, you can change the position of that field.

The Yank menu option is automatically canceled when you have selected all the available statistic fields on any given screen or when no more fields are available on the Custom Statistics screen.

Using the yank-and-put method is the recommended approach for creating the “Custom Statistics” screen. You browse through the existing screens and select the set of statistic fields you want to monitor.

MANUAL CUSTOM
 SCREEN DESIGN

If you are an advanced user, you can also manually create a statistic field. On the “Custom Statistics” screen, you can use the Config menu option to explicitly enter the index of the specific statistic field, numbered from 1 to 1020. On OpenVMS, these indexes are documented in the following text file:

SYSS\$LIBRARY:RMU\$SHOW_STATISTICS.CDO

If you have installed multiple versions of Oracle Rdb on your system, the text file is named

SYSS\$LIBRARY:RMU\$SHOW_STATISTICS w .CDO

where the w is the Oracle Rdb software version.

When you select statistic fields manually, the keyboard will beep if you specify the index of an existing statistic field. However, the RMU Show Statistic utility will not display an error message.

CUSTOM SCREEN
 CUSTOMIZATION

When you type “**C**” on the “Custom Statistics” screen, the RMU Show Statistic utility will display the configuration menu. The configuration menu allows you to add, delete, move, compress, and annotate statistic fields on the “Custom Statistics” screen. The configuration menu varies according to the state of the “Custom Statistics” screen but typically all five options are displayed.

Add a statistics field When you select the add menu option, you are prompted to select the screen position where you want to place the statistic field, the index of the statistic field, and the title of the statistic field. The title you specify is then appended to the respective index, in brackets, to show which statistics have been manually selected.

This option is extremely useful for tracking statistics that are not normally displayed. For instance, the statistic field that tracks the total number of bytes of virtual memory allocated is not displayed by any RMU Show Statistic utility screen. However, you can select index number 16 and the number of bytes of virtual memory allocated is displayed on the “Custom Statistics” screen.

Delete a statistic field When you select the delete menu option, you are prompted to select the statistic field you want to delete.

Move a statistic field When you select the move menu option, you are prompted to select the statistic field to be moved and the screen position where the selected field will be displayed.

Annotate a statistics field Selecting the annotate menu option allows you to change the name of the yanked statistic field. This is useful when the yanked field name is no longer unique on the “Custom Statistics” screen.

Note that if you export a configuration file that contains an annotated statistic field name, subsequently importing the configuration file will not identify the annotated name as a valid statistic field name.

Compress the display Selecting the compress menu option eliminates all intervening blanks lines on the screen. This option is particularly useful after you have deleted one or more statistic fields.

Initially, the “Custom Statistics” screen appears as follows:

CUSTOM SCREEN
APPEARANCE

```

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds Custom Statistics Elapsed: 00:00:21.23
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
database binds          0          0          0.0          1          0.0
transactions            0          0          0.0          0          0.0
-----
Config Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write

```

The initial “database binds” and “transactions” statistic fields are already stored on the “Custom Statistics” screen for you. You may delete them if you wish; the method to accomplish this will be explained a little later in this section.

Adding Screen Fields To yank one or more fields from the “VM Usage Statistics” screen to the “Custom Statistics” screen, proceed to the “VM Usage Statistics” screen:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds VM Usage Statistics Elapsed: 00:24:35.29
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
```

```
-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
GET_VM calls      187      0      0.2      351      0.0
FREE_VM calls      0      0      0.0      14      0.0
GET_VM kilobytes   795      0      1.6      2402     0.0
FREE_VM kilobytes  32      0      0.7      978     0.0
$EXPREG calls      0      0      0.0      0      0.0
-----
```

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

Type “Y” to display the Yank-and-Put menu:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds VM Usage Statistics Elapsed: 00:00:04.65
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
```

```
-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
A. GET_VM calls    74      0      72.9     339     0.0
B. FREE_VM calls   0      0      0.4      2      0.0
C. GET_VM kilobytes 343     0      336.3    1564    0.0
D. FREE_VM kilobytes 31     0      30.1     140     0.0
E. $EXPREG calls    0      0      0.0      0      0.0
-----
```

Type <return> or <letter> to select stats field, <control-Z> to cancel

Type “A” to select the GET_VM calls statistic field. You remain in the selection mode but the selection menu changes with each selection. After you select option “A”, the new menu is displayed as follows:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds VM Usage Statistics Elapsed: 00:00:04.65
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
```

```
-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
GET_VM calls      74      0      72.9     339     0.0
A. FREE_VM calls   0      0      0.4      2      0.0
B. GET_VM kilobytes 343     0      336.3    1564    0.0
C. FREE_VM kilobytes 31     0      30.1     140     0.0
D. $EXPREG calls    0      0      0.0      0      0.0
-----
```

Type <return> <letter> to select stats field, <control-Z> to cancel

Because you have already selected the GET_VM calls statistic field, you cannot select it again, so it has been removed from the selection menu. To select the GET_VM

kilobytes field, type **"B"**. Press Ctrl/Z to exit the "VM Usage Statistics" screen yank-and-put menu.

Now, return to the "Custom Statistics" screen. The GET_VM calls and GET_VM kilobytes fields that you selected with the yank-and-put method are displayed:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds Custom Statistics Elapsed: 00:05:13.90
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
-----
statistic..... rate.per.second..... total..... average.....
name..... max..... cur..... avg..... count..... per.trans....
database binds 0 0 0.0 1 0.0
transactions 0 0 0.0 0 0.0
GET_VM calls 74 0 1.0 341 0.0
GET_VM kilobytes 343 0 5.4 1704 0.0
-----
Config Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write
```

Note the statistic fields selected using the yank-and-put method do not have the index notation appended to the field name.

Deleting Custom Screen Fields If you do not want to display these statistic fields, type **"C"** to choose the Config menu option. This displays the configuration menu:

```
Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
Rate: 3.00 Seconds Custom Statistics Elapsed: 00:09:29.95
Page: 1 of 1 DISK$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online
-----
statistic..... rate.per.second..... total..... average.....
name..... max..... cur..... avg..... count..... per.trans....
database binds 0 0 0.0 1 0.0
transactions 0 0 0.0 0 0.0
GET_VM calls +---Select Custom Configuration---+ 343 0.0
GET_VM kilobytes | A. Add a statistic field | 1843 0.0
| B. Delete a statistic field |
| C. Move a statistic field |
| D. Annotate a statistics field |
| E. Compress the display |
+-----+
-----
Type <return> or <letter> to select customization choice, <control-Z> to cancel
```

To delete an existing statistic field, choose menu option **"B"**. The list of existing statistic fields is displayed and you can select the field to be deleted.

Once the configuration operation is complete, the "Custom Statistics" screen is displayed again:

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 18-JAN-1998 12:04:26.62
 Rate: 3.00 Seconds Custom Statistics Elapsed: 00:29:59.74
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.STATS]MF_PERSONNEL.RDB;1 Mode: Online

```
-----
```

statistic.....	rate.per.second.....			total.....	ave rage.....	
name.....	max.....	cur.....	avg.....	count.....	per.trans....	
database binds	0	0	0.0		1	0.0
transactions	0	0	0.0		0	0.0
GET_VM kilobytes	343	0	1.5	2682		0.0

```
-----
```

Config Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write

Note that the row where the deleted statistic field appeared is now blank.

Compressing Screen Fields The display is not automatically compressed to eliminate empty lines. Press “**C**” from the on-screen menu to select the Config option, then select “**D**” from the Custom Configuration menu to compress (eliminate) the empty lines.



You can save your “Custom Screen” configuration by selecting the “Export Configuration File” option in the Tools menu. However, if you annotated your custom screen statistic field names, the annotated field names cannot be exported.

Log Files

The RMU Show Statistic utility is able to produce a number of log files. These log files contain textual information so that you can view them later using your favorite display utility or editor.

The log files are not available if the INPUT qualifier is specified.

You do not need to display the corresponding screen to record to the log file. The log files are maintained regardless of which screen, if any, is displayed.

You can have any or all log files active at the same time.

The log file information is written at the current screen refresh rate. Using a larger screen refresh rate minimizes the size of the file, but results in a large number of missed messages. Using a smaller screen refresh rate produces a large log file, but contains a much finer granularity of messages.

By default, the log files are not written.

You can also enable and disable the log files at runtime using the Tools menu, which you can obtain using the exclamation point (!).

Starting with Oracle Rdb 7.0.2.0, you can view the log files while the RMU Show Statistic utility is active; previously, you needed to stop the logging before the log files could be accessed. However, for performance considerations, the RMU Show Statistic utility does not flush the log file contents too often, so viewing the log files while they are being written is slightly delayed.

STALL MESSAGE
LOG FILE



Writing stall messages to a log file can be useful when you notice a great number of stall messages being generated, but do not have the resources to immediately investigate and resolve the problem. The log file generated can be reviewed later so that you can trace the problems and solve them.

The advantage of using a stall log is that the lock stalls automatically display the lock conflict information, which is an absolutely vital piece of information when examining the log files after the fact.

Appendix J contains information regarding how to send stall message information to an OpenVMS mailbox using the RMU Show Statistic utility “Stall Message Logging” facility. This allows you to trap stall messages written in conjunction with the ALARM qualifier, and act upon these messages using a user-defined procedure.

The following is an example of a stall message log file:

```
Oracle Rdb X7.0-00 RMU Show Statistic utility Stall Log Database
DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1
Stall Log created 26-JAN-1998 10:03:44.07
10:04:22.81 38006AE8:1 10:04:22.78 writing 222 pages back to dat abase
10:13:49.45 3800A2AF:1 10:13:49.51 reading pages 1:1323 to 1:1325
10:13:51.61 3800B4AD:1 10:13:51.68 waiting for record 77:97:-1 (CR)
  State... Process.ID Process.name... Lock.ID. Rq Gr Queue "record 77:97:-1"
  Owner: 3800B4AD RICK6 29000719 CR CR Grant
  Owner: 3800B4AC RICK5 3F001829 EX CW Grant
10:13:51.61 3800B4B0:1 10:13:51.68 writing RUJ file block 1
10:13:52.69 3800B4AC:1 10:13:52.33 waiting for record 59:486:0 (EX)
  State... Process.ID Process.name... Lock.ID. Rq Gr Queue "record 59:486:0"
  Blocker: 3800B4AD RICK6 2C00BCEB EX PR Cnvrt
  Waiting: 3800B4AC RICK5 080096C9 EX PR Cnvrt
  Waiting: 3800A2AF RICK7 730096A9 PR NL Wait
10:13:52.69 3800B4AD:1 10:13:52.31 waiting for record 59:486:0 (EX)
  State... Process.ID Process.name... Lock.ID. Rq Gr Queue "record 59:486:0"
  Waiting: 3800B4AD RICK6 2C00BCEB EX PR Cnvrt
  Waiting: 3800B4AC RICK5 080096C9 EX PR Cnvrt
  Waiting: 3800A2AF RICK7 730096A9 PR NL Wait
```

LOCK TIMEOUT
LOG FILE

Writing lock timeout messages to a log file can be useful when you notice a great number of lock timeouts being generated, but do not have the resources to immediately investigate and resolve the problem. The log file generated can be reviewed later so that you can trace the problems and solve them.



Lock timeouts are considerably rarer than lock deadlocks, so they should be reviewed more closely.

The following is an example of a lock timeout log file:

```
Oracle Rdb X7.0-00 RMU Show Statistic utility Lock Timeout Log Database
DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1
Lock Timeout Log created 26-JAN-1998 11:53:20.28
3800B06E:2u 12:39:22.46 - waiting for quiet (EX)
```

LOCK DEADLOCK
LOG FILE

Writing lock deadlock messages to a log file can be useful when you notice a great number of lock deadlocks being generated, but do not have the resources to immediately investigate and resolve the problem. The log file generated can be reviewed later so that you can trace the problems and solve them.



Note that the OpenVMS "DEADLOCK_WAIT" SYSGEN parameter directly affects lock deadlock problems and their occurrence rate.

The following is an example of a lock deadlock log file:

```
Oracle Rdb X7.0-00 RMU Show Statistic utility Lock Deadlock Log Database
DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1
Lock Deadlock Log created 26-JAN-1998 11:37:41.28
3800B4AC:1 11:37:38.73 - waiting for page 5:457 (PW) [444 missed]
3800B4AD:1 11:37:45.97 - locking page 5:965 [418 missed]
3800A2AF:1 11:37:05.17 - waiting for page 5:1141 (PR) [438 missed]
3800B4B0:1 11:37:34.72 - waiting for page 5:1262 (PR) [446 missed]
3800B4AD:1 11:37:50.57 - waiting for page 5:470 (PW)
3800B4B0:1 11:37:51.81 - locking page 34:309
3800A2AF:1 11:37:51.95 - locking page 1:487
3800A2AF:1 11:37:58.14 - waiting for page 5:714 (PW)
3800A2AF:1 11:38:05.22 - waiting for page 5:860 (PR)
3800B4B0:1 11:38:09.12 - locking page 5:1028
3800B4AC:1 11:38:17.37 - waiting for page 5:532 (CR)
3800A2AF:1 11:38:38.68 - waiting for page 5:842 (PR)
3800B4AD:1 11:38:42.35 - waiting for page 5:1268 (PR)
3800B4B0:1 11:38:44.34 - locking page 5:612
3800B4B0:1 11:38:54.32 - locking page 5:483
3800B4AC:1 11:38:54.97 - waiting for page 5:1210 (PR)
3800A2AF:1 11:38:57.97 - waiting for page 5:994 (PW)
3800B4AC:1 11:39:10.81 - locking page 5:1330
3800B4AD:1 11:39:16.77 - waiting for page 5:1268 (PR)
```

DBKEY ACCESS
LOG FILE

Logs the records accessed during a given processing period by the various attached processes. You are prompted to enter a file name where all the accessed DBKEYs are logged.

The following is an example of a DBKEY access log file:

```
Oracle Rdb X7.0-00 RMU Show Statistic utility DBKEY Log Database
DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1
DBKEY Log created 26-JAN-1998 08:12:01.18
Process.ID      CurrentTime Data.DBKEY  Snap.DBKEY  SPAM.DBKEY  AIP.DBKEY  ABM.DBKEY
3800B4AD:1      10:18:53.21 1:7         34:828      1:1         1:464      1:479
3800A2AF:1      10:18:53.21 5:112       30:9         1:1         1:464      1:479
3800B4B0:1      10:18:53.21 5:363       30:7         1:1         1:464      1:479
3800B4AC:1      10:18:54.27 1:7         30:19        1:1         1:464      1:479
3800A2AF:1      10:18:54.27 1:6         30:9         1:1         1:464      1:479
3800B4B0:1      10:18:54.27 1:6         34:808       1:1         1:464      1:479
3800B4AC:1      10:18:56.39 1:7         34:830       1:1         1:464      1:479
3800B4AD:1      10:18:56.39 5:193       34:828       1:1         1:464      1:479
3800A2AF:1      10:18:56.39 1:6         34:803       1:1         1:464      1:479
3800B4B0:1      10:18:56.39 1:6         30:8         1:1         1:464      1:479
3800A2AF:1      10:18:57.46 1:976       30:19        1:1         1:464      1:479
3800B4B0:1      10:18:57.46 5:47        30:9         1:1         1:464      1:479
3800B4AC:1      10:18:58.52 5:33        30:24        1:1         1:464      1:479
3800B4AD:1      10:18:58.52 5:120       30:21        1:1         1:464      1:479
3800A2AF:1      10:18:58.52 5:401       30:6         1:1         1:464      1:479
3800B4B0:1      10:18:58.52 1:976       30:6         1:1         1:464      1:479
3800B4AC:1      10:18:58.86 1:794       30:24        1:1         1:464      1:479
3800B4AD:1      10:18:58.86 1:794       30:21        1:1         1:464      1:479
3800A2AF:1      10:18:58.86 5:23        30:4         1:1         1:464      1:479
3800B4B0:1      10:18:58.86 1:976       30:8         1:1         1:464      1:479
```




Note that it is not possible to record every DBKEY accessed. Only the last DBKEY accessed at the refresh interval is written to the log file. Therefore, you may need to use a faster refresh rate when recording the DBKEY log file.

Process Identifier Suffixes

On screens displaying process information, each process is identified using the following format:

process identifier : stream identifier [type tag]

The process identifier is assigned by the operating system and uniquely identifies the process.

The stream identifier is assigned by Oracle Rdb and together with the process identifier uniquely identifies an attached process; a single operating system process may have multiple database attaches.

The optional type tag is displayed by the RMU Show Statistic utility to provide additional information about the type of process. The following type tags are displayed:

<u>Type Tag</u>	<u>Description</u>
(blank)	Ordinary application process
D	Process is being recovered by a database recovery (DBR) process
R	Process is a remote-connection server
s	Process is a database server
u	Process is a database utility
*	Process is attached on a remote node
A	Process is attached by the RMU Show Statistic utility for global statistic collection
G	Process is activated for global statistic collection but is not attached

Configuration File Management

Chapter 5 - Configuration File Management

INTRODUCTION

The RMU Show Statistic utility provides the ability to import and export a configuration file at runtime. The configuration file is a human-readable file that you create using the editor of your choice. You can also automatically generate a configuration file from the RMU Show Statistic utility using the current runtime utility configuration settings as the configuration file default values.

The RMU Show Statistic utility processes the configuration file prior to opening the database or the binary input file, if you specify the optional `CONFIGURE=config_file name` qualifier. If you do not specify the `CONFIGURE` qualifier, all of the variables default to values based on command-line qualifiers and logical names. The default configuration file type is `.CFG`.

The configuration file is processed in two passes. The first pass occurs before the database is opened and processes most of the configuration file entries. The second pass occurs after the database is open, and processes those variables that are database dependent, such as the “`CUSTOM_LINE_n`” variables.

CREATING
 CONFIGURATION
 FILES

The RMU Show Statistic utility configuration file is a human-readable file, created using the editor of your choice. The configuration file typically resides in the database directory, although it can reside anywhere the DBA desires.

You can also manually export a new configuration file by selecting the Tools menu, which you obtain using the exclamation point (!), and then selecting the “Save current configuration” option. You will be prompted to enter the name of the new configuration file. The default configuration file type is `.CFG`. For portability, comments are generated using the pound sign (#) character. The configuration file will be generated using the current settings as the configuration variable values.

Each entry in the configuration file uses the following general format:

```
variable=value;
```

The equal sign (=) separating the variable and the value is required. Also note that each definition is terminated using the semicolon (;).

You can specify a comment using either the pound sign (#) or exclamation point (!) characters, and continues to the end of the current line. A comment can occur anywhere in a line, but always terminates that line. Blank lines are ignored.

The “variable=value;” syntax is free-format. That is, spacing is not relevant to the parsing of the token. For example, the entry

```
STALL_LOG=“STALL.LOG”;
```

could be entered in the configuration file as:

```
STALL_LOG
=
“STALL.LOG”
;
```

However, for readability reasons, this type of formatting is not recommended. Also, multiple entries can be put on a single line, although this is also not recommended, for readability reasons. For example, the following entry is permitted:

```
STALL_LOG=“STALL.LOG”; DBKEY_LOG=“DBKEY.LOG”;
```

Really long configuration file entries can be “continued” on the next line by terminating the line with a back-slash (\). Lines can be “continued” practically indefinitely, up to 2048 characters, even within quoted values; for example:

```
EVENT_DESCRIPTION=“ENABLE 'pages checked'
MAX_CUR_TOTAL \
INITIAL 7 \
EVERY 11 \
LIMIT 100 \
INVOKE DB_ALERT”;
```

The line-continuation feature is not limited to the EVENT_DESCRIPTION variable; it can be used for any configuration variable. As an added bonus, comments can be embedded in continued lines if they start at the beginning of the next line. For example, consider the following two event descriptions:

```
EVENT_DESCRIPTION=“ENABLE ' (Asynch. reads)'
MAX_CUR_TOTAL \
! this will work as expected AREA EMPIDS_OVER \
INITIAL 6 EVERY 10 LIMIT 100 INVOKE DB_ALERT”;
```

```
EVENT_DESCRIPTION="ENABLE ' (Asynch. reads)'  
MAX_CUR_TOTAL \  
AREA EMPIDS_OVER ! this will NOT work as expected \  
INITIAL 6 EVERY 10 LIMIT 100 INVOKE DB_ALERT";
```



Note that the comment in the second event description takes precedence over the line continuation character.

IMPORTING
CONFIGURATION
FILES

A new configuration file can be imported at any time by selecting the Tools menu, which you can obtain using the exclamation point (!), and then selecting the “Import configuration settings” option. You will then be prompted to enter the name of the new configuration file. The default configuration file type is .CFG.

NESTED
CONFIGURATION
FILES

A nice feature of configuration files is that they can also be nested to any depth, using the special “INCLUDE” variable. This command variable will be described in detail below. The use of nested configuration files allows you to create hierarchies of configuration variable definitions.

When using nested configuration files, remember that the last definition of a variable is the one that is available for use by the RMU Show Statistic utility.

The use of nested configuration files is a good method for “grouping” related configuration file objects, such as events, into their own configuration file.



Note that, when including other configuration files, error messages for those included configuration files are displayed in the corresponding log file. For example, if the configuration file CONFIG.CFG includes the configuration file EVENTS.CFG, any errors or log messages pertaining to the EVENTS.CFG configuration file will be placed in the EVENTS.LOG log file, while all other log messages will be placed in the CONFIG.LOG log file.

Configuration Variable Semantics

The “variable” is either a pre-defined configuration parameter known to the RMU Show Statistic utility, or a user-defined variable whose value will presumably be used later as the value of some other pre-defined configuration parameter. The complete list of pre-defined variables is described later in this document.

All variable names must start with an alphabetic character. The “variable” name is case-insensitive.

All variables known to the RMU Show Statistic utility have defaults; these defaults are described later in this document. It is not necessary to explicitly specify every variable. It is generally recommended that you only explicitly specify those variables whose default value is not acceptable to your application. Most configuration file variables replace existing command-line qualifiers or logical names.

VARIABLE TYPES

Configuration variables have seven types. These types dictate how the corresponding atomic value is available for use. The variable types are:

- Numeric This is the most common type of configuration variable. Some, but not all, numeric variables have minimum and maximum values. Also, some, but not all, numeric variables have scale values, which dictate the number of fractional digits allowed. For example, a variable with a scale of “2” can be specified in terms of hundredths. A scale of “0” designates only a whole number is allowed (0 decimal positions).
- Boolean These types of variables can use the values TRUE and FALSE, or ENABLED and DISABLED interchangeably. These values are not quoted. You can also use the numeric values “0” and “1”.
- String String variables specify a quoted value, which are typically file names. Null string values are specified using an empty set of quotes (“”).
- Date Date variables specify a quoted value in the standard operating system date format.
- Command Command variables cause an action specified by the quoted value to be performed. The most common variable is the “PRINT”, “PROMPT” and “INQUIRE” variables. Command variables are not automatically exported.
- Control Control variables are most commonly used to document interesting values. They do not actually do anything, but they can be useful as the value of another variable. The most common variable is the “DATABASE” variable.
- User Defined User defined variables are used as the values of other types of variables, and are treated as simple strings. User defined variables are not exported.

Configuration Value Semantics

The value is the numeric or quoted-string value to be assigned to the respective variable. The value, when quoted, is case-sensitive.

The values for numeric variables can be specified using either whole numbers or floating-point numbers, depending on the respective “scale” of the particular variable. The scale of a variable indicates the number of digits to the right of the decimal place.

Obviously, if you specify a value whose fractional portion exceeds the scale, then the value will be truncated. For example, if you specify “4.25” for a variable whose scale is “1” (meaning 1 digit to the right of the decimal point) then only “4.2” would be used. All values are truncated according to the specified scale.

QUOTED VARIABLE
EVALUATION

When a variable's value is quoted using double quotes (“ ”), the evaluation of the value is treated as a hierarchy of values. The evaluation hierarchy is: variable name, logical name, atomic value. This hierarchy means that every value is first iteratively treated as another variable, then iteratively treated as a logical name, then finally as an atomic value.

Note that if you can also specify a single-word string value without the double quotes, if so desired. For example, the value “FOOBAR” could be specified using either double quotes or no quotes at all, since “FOOBAR” is a single-word string. However, the value “here and now” must be enclosed with either single- or double quotes.

Quotes can also be nested, as long as they alternate. Note the pairings of the quotation marks in the following example:

```
QUOTE_TEST = "1 '2 "3 '4 "5 '6 "7 '8 8' 7" 6' 5" 4' 3" 2' 1";
```

HIERARCHICAL
VARIABLE
EVALUATION

The hierarchical evaluation semantics allows you to specify the value of one variable as the value of another.

In the following example, the user-defined variable TEST is defined with the value “0”. The pre-defined numeric variable CYCLE is defined to the value “TEST”, which because it is a variable will be evaluated as the value “0”. Finally, the pre-defined com-

mand variable PRINT will be evaluated as “CYCLE”, which because it is also a variable, will be evaluated as the value “0”.

```
TEST = 0;
CYCLE = “TEST”;
PRINT = “CYCLE”;
```

It is interesting to note that, even though the CYCLE variable is of type “numeric”, you can specify a quoted string value as long as it ultimately evaluates to a numeric value.

It is also interesting to note that the hierarchical evaluation of values prevents a quoted value from being the same as the variable name. For instance, in the following configuration file entry the value “INCLUDE” obviously cannot be evaluated as the variable name “INCLUDE”, since this would be recursive.

```
INCLUDE=“INCLUDE”;
```

LOGICAL NAME
EVALUATION

Using logical names as the value of variables is very useful for certain variables, such as the INCLUDE variable (see section “Nested Configuration Files”). This allows users to customize their configuration files through the use of logical names while still utilizing a centralized configuration file hierarchy.

VARIABLE
KEYWORD
EVALUATION

Some variables have special keywords for their values. These keywords must be specified in upper-case, exactly as they are described later in this document. Be especially careful when specifying user-defined variables whose name conflicts with any of the valid keywords. When using atomic values for keywords, it is recommended to use single quotes to avoid inadvertent evaluation of the keyword as another variable.

When a variable's value is quoted using single quotes ('), the value is treated as a simple string. For more information and examples on the difference between single quoted and double quoted strings, refer to the “Print Variable” section below.

PRINT & PROMPT
COMMAND
VARIABLES

The PRINT and PROMPT variables are useful for debugging and tracing execution of the configuration file. For instance, the following example demonstrates how to display a variable's default value, and then the new value initialized from the configuration file:

```
PRINT=“CYCLE“;
CYCLE=5;
PRINT=“CYCLE”;
```

The output from the previous example is:

```
# This file was generated by the Show Statistic utility.
# Generated on 16-APR-1997 10:35:05.48
line 3: variable "PRINT" value "CYCLE = 0"
line 5: variable "PRINT" value "CYCLE = 5"
```

The PRINT variable is also useful for tracing execution of the configuration file, which may be very useful when using nested configuration files. For example, the PRINT command can be used to display simple messages to the log file:

```
PRINT = "Now initializing CYCLE variable";
CYCLE = 5;
```

The output from the previous example is:

```
# This file was generated by the Show Statistic utility.
# Generated on 16-APR-1997 10:35:05.48
line 1: variable "PRINT" value "Now initializing CYCLE variable"
```

Care should be taken when printing single-word messages, as these may be evaluated as variables. Consider the following example:

```
HERE = "there";
PRINT = "here";
PRINT = 'here';
PRINT = "here and there";
```

In this example, the value of the first PRINT variable is "there" not "here", because "here" is a user-defined variable. The value of the second PRINT variable is "here" because it is a single quoted simple string. Finally, the value of the third PRINT variable is "here and there" because it is an atomic string.



Note that the PROMPT variable does not work during an import operation.

VARIABLE VALUE
 REDIRECTION

Redirecting is an interesting use of a variable, especially a variable whose value is input by the user. Consider the following configuration file example:

```
OVER = "AND OUT";
THERE = "OVER";
HELLO = "THERE";
FOO = "HELLO";
```

Setting the variable "FOO" to the value "HELLO" causes an implicit redirection of its value to be the final value of all of its own value(s). This results in the "FOO" variable being set to the value "AND OUT", which is the final value of all values.

The REDIRECT command variable is extremely useful following an INQUIRE command, where the user entered the name of another variable to use. By redirecting the variable's value, you can essentially implement a menu mechanism.

For example, consider the following configuration file example:

```
VALUE1 = "10";
VALUE2 = "20";
VALUE3 = "30";
PROMPT = "Enter VALUE1, VALUE2 or VALUE3";
INQUIRE = "VALUE";
REDIRECT = "VALUE";
PRINT = "VALUE";
```

Note how this differs from setting the VALUE variable to itself.

Log File

If any problem occurs processing the specified configuration file, a log file is automatically created. The log file's name is the same as the configuration file name with the .CFG file type replaced with a .LOG file type. For example, if the configuration file name is "CONFIG.CFG" then the corresponding log file name would be "CONFIG.LOG". Log file entries have the following format:

"line #: message".

The following example shows a problem trying to include a nested configuration file; the offending command is on line 4 of the configuration file.

```
# This file was generated by the Show Statistic utility.
# Generated on 6-APR-1997 07:20:00.73
line 4: INCLUDE="config.cfg";
line 4: command "INCLUDE" failed %COSI-E-FLK, file currently locked by
another user
```

You can also specify the LOG qualifier, which lists each processed variable to the corresponding log file. This qualifier is highly recommended until you become more familiar with using the configuration files.

Note that, when including other configuration files, error messages for those included configuration files are displayed in the corresponding log file. For example, if the configuration file CONFIG.CFG includes the configuration file EVENTS.CFG, any errors or log messages pertaining to the EVENTS.CFG configuration file will be placed in the EVENTS.LOG log file, while all other log messages will be placed in the CONFIG.LOG log file.

Example of A Really Bad Configuration File

The following is an example of a really bad configuration file. It is presented here to provide examples of common mistakes to avoid.

Each configuration file entry below demonstrates a common problem encountered using configuration variables. The problem or expected result is described as the entry comment.

```

PRINT = "DATABASE";           # display current database name
PRINT = 'DATABASE';          # should display "database"
UNSET = "FOO";               # warning: unknown variable "FOO"
FOO = 5;
UNSET = 'FOO';               # warning: unknown variable "FOO" (not variable)
UNSET = "FOO";
PRINT = "FOO";               # should display "FOO" instead of "5"
123 = 456;                   # warning: incorrect variable name "123"
PRINT = "PRINT";            # should display "PRINT"
INCLUDE = "INCLUDE";        # %COSI-E-FNF, file not found
INCLUDE = "CONFIG.CFG";     # %COSI-F-EEXIST, file exists
CUSTOM_LINE_5 = "transactions # warning: unterminated string (expecting " or ')";
CUSTOM_LINE_5 = "transactions";
CUSTOM_LINE_5 = "verb successes"; # warning: replacing defined variable "CUSTOM_LINE_5"
CUSTOM_LINE_6 = "verb successes"; # warning: replacing defined statistic "transactions"
CUSTOM_LINE_7 = "verb failures";
CUSTOM_LINE_8 = "verb confusion"; # warning: custom statistic field "verb confusion" not
found
HERE = "there";
PRINT = "here and now";     # should display "here and now"
PRINT = "here";             # should display "there"
PRINT = 'here';             # should display "here"
YES = "yes";
PRINT = "yes";              # should display "YES = yes"
PRINT = "CYCLE";           # should display "CYCLE = 0"
CYCLE = "5";                # this is reasonable
CYCLE = 5;                  # this also
PRINT = "CYCLE";           # should display "CYCLE = 5"
CYCLE = ALARM;
PRINT = "CYCLE";           # should display "CYCLE = 0"
NO = "no";
PRINT = "no";               # should display "NO = no"
BROADCAST = BROADCAST;     # warning: invalid value "BROADCAST"
NOTIFY = "CLUSTER";        # should display "NOTIFY = ""

```

Cluster Statistic Collection

Chapter 6 - Cluster Statistic Collection

INTRODUCTION

The ability to support the expanding roles of data warehousing and OLTP systems requires ever-increasing numbers of disk devices and additional processing resources. Cluster technology is one method by which DBAs can scale the application and database in support of these types of systems. While clusters provide additional processing capabilities, they also have increased overhead at the disk, locking, and network layers. Therefore, it is not always possible to achieve a completely linear increase in performance by simply scaling the application using a clustered system. Clusterwide statistical monitoring and measurement tools are absolutely essential to quantitatively determine the “cost” of clustering the database.

Most database products provide tools for the purpose of collecting database statistics and monitoring the performance characteristics of the application. However, these tools seldom provide clusterwide statistical information collection capability, which is absolutely critical to the analysis of performance problems. Furthermore, these tools seldom provide analytical tools for comparing single-node access to cluster access.

The Oracle Rdb 7.0.1.0 product includes the “Cluster Statistics Collection and Presentation” facility as an enhancement to the RMU Show Statistic utility. The purpose of this facility is to provide for the realtime collection of database statistics from other nodes within the cluster where the database is currently being accessed. These statistics include not only summary numeric information, but also “Stall Messages” and drill-down information such as per-storage area information.

The “Cluster Statistics Collection and Presentation” facility operates on the single database the RMU Show Statistic utility is currently accessing. You cannot collect statistics for multiple databases simultaneously with a single RMU Show Statistic session. However, you can switch between multiple databases in a serial manner.

COMMAND
 QUALIFIER

The “Cluster Statistics Collection and Presentation” facility is available using the RMU Show Statistic utility CLUSTER command qualifier.¹⁴ This qualifier optionally specifies a list of cluster nodes from which to perform statistic collection.

The following example demonstrates the use of the CLUSTER qualifier to initiate statistic collection from two cluster nodes:

```
$ RMU /Show Statistic -
_$ /CLUSTER=(MYNODE, ALPHA4) -
_$ MF_PERSONNEL
```

When the user does not specify the CLUSTER qualifier, or specifies the NOCLUSTER qualifier (the default), cluster statistics will not be active. However, the user may still enable clusterwide statistic collection online using the Tools menu. A discussion of the online maintenance of cluster statistic collection nodes occurs later.

CONFIGURATION
 VARIABLE
 SUPPORT

If you are using a configuration file, you can specify the CLUSTER_NODES configuration variable instead of the CLUSTER command qualifier. This variable has all of the same semantics of the command qualifier, but is more persistent because of its explicit association with the database.

The following example demonstrates the use of the CLUSTER_NODES configuration variable as it would appear in a configuration file:

```
CLUSTER_NODES="MYNODE,ALPHA4";
```

The keyword ALL_OPEN indicates that cluster statistic collection is to occur on all nodes where the database is currently open.

STATISTIC
 COLLECTION SET

The complete set of nodes for which the statistic collection facility is active on the database is the collection set. A collection set can be either implicit, meaning statistic collection occurs on all open nodes, or explicit meaning the user specifies a collection set.

Note that the collection set differs from the set of open database nodes, in that not all open database nodes have to participate in the collection set.

If you specify the CLUSTER qualifier by itself, then cluster statistic collection will occur on all cluster nodes for which the database is currently open. This defines the *implicit* collection set. If you specify the CLUSTER=(*node_list*) qualifier, then the facility

¹⁴ The “Cluster Statistics Collection and Presentation” facility is presently only available on OpenVMS platforms

will collect cluster statistic information on the specified nodes only, even if the database is not yet open on those nodes. This defines the *explicit* collection set.

The difference between an *implicit* and *explicit* collection set is important when switching databases from within the RMU Show Statistic utility. Please refer to the “Switching Databases” section for more information on the relevant differences.

CLUSTER NODE
ACCESS

Up to 95 different cluster nodes can be specified using the CLUSTER qualifier. There is a maximum number of 95 cluster nodes because Oracle Rdb currently supports a maximum number of 96 nodes per database. The current node is always an implicit member of the statistic collection set.

The node name specified can also be a logical name whose definition is a comma-separated list of cluster nodes. A problem that commonly occurs with comma-separated values is to treat the logical name as a search list. Since the cluster collection facility does operate with search list logical names, it is therefore necessary to place the comma-separated cluster node list in quotation marks. For example, the following defines a logical name containing the names of three cluster nodes:

```
$ DEFINE NODE NAME "MYNODE,ALPHA4,ALPHA5"
```

It is important to use the node name logical name instead of a cluster alias. Cluster aliases do not expand into their component node names. Consequently, cluster aliases cause confusion in the identification of which nodes of the cluster are actively collecting statistics.

You do not have to have the RMU Show Statistic utility running on any of the specified cluster nodes. The database does not even have to be open on the other cluster nodes; the facility is capable of opening the database automatically.

SWITCHING
DATABASES

Implicit and explicit collection sets are very important when using the “Switch Database [→” option of the RMU Show Statistic utility. If you specify the CLUSTER qualifier with a node list, that list of cluster nodes will apply to any and all databases accessed during the statistic collection session. However, if you specify the CLUSTER qualifier without a node list, then the collection set will contain only those nodes actually open for the database.

If you manually add a node using the “Cluster Statistics” menu, which is located in the “Tools” submenu, the specified node will not be a member of a new database’s cluster collection set. In other words, manually adding and deleting cluster nodes only affects the current database, and does not apply to any other database accessed during the session.

For example, assume you are running the RMU Show Statistic utility on database DB1 on node MYNODE with the manually added cluster node MYNODE. Subsequently switching to database DB2 as the new current database will not display cluster statistics from node MYNODE unless you manually add that node. Furthermore, switching back to database DB1 as the new current database loses the previous collection of node MYNODE because it was part of the original collection set.

There is no method available to exclude the current node from the current collection set. You must collect statistics from another node if you want to accomplish this.

If the database is not currently open on a node of the specified cluster node list, the facility will still attempt to perform cluster statistic collection. However, it will still be necessary for the DBA to open the database normally prior to having regular application process attaches. This means that the Cluster Statistic Collection facility will not compromise the security of databases whose “open mode” is MANUAL.

HOW CLUSTER
 STATISTICS
 COLLECTION
 WORKS

The “Cluster Statistics Collection and Presentation” facility uses a special statistic collection server on each of the specified remote collection nodes. This statistic collection server is a lightweight process that accesses the database global section on that node, and collects the requested statistic information. The exchange of information with the primary node uses either the DECnet or TCP/IP network protocols.

The collection of cluster statistics occurs at the specified display refresh rate. Therefore, it is important that you set the display refresh rate to a reasonable rate based on the number of cluster nodes being collected. The default refresh rate of three seconds is reasonable for most cluster collection loads. A refresh rate of one second is useful when detailed problem analysis is necessary to produce the finer granularity statistical information; however, you should not sustain a one-second refresh rate for long periods of time.

Collection of the main set of “summary” numeric statistic information always occurs when using the cluster statistic collection facility. However, the cluster collection of the by-cache, by-area and by-logical-area statistics occurs depending upon the current screen being displayed.

For example, the collection of the “per-process” stall information occurs only while viewing one of the stall messages screens. In addition, the collection of statistics does not occur for all screens; the facility will notify you when you migrate to a screen that does not support cluster statistic collection. However, the majority of statistic screens support cluster statistic collection.

While the cluster statistic collection and presentation facility is active, all on-screen menu options continue to operate normally. This includes the time-plot screens, scatter-plot screens, screen pause and various other options.

ENABLED VERSUS
 ACTIVE NODE
 STATUS

When you specify a list of nodes, either implicitly or explicitly, these nodes are part of the “enabled” members of the collection set. However, the members are either active or inactive, depending on their statistic collection status. An inactive node is one that is inaccessible, while an accessible node is active.

Inactive nodes are not immediately removed from the collection set. The DBA has the option of manually removing the inactive node from the collection set, or indicating that the inactive node should be re-connected if possible. Failure to re-connect to an inactive node will result in automatic removal of the inactive node from collection set.

NETWORK
 PROTOCOLS

The “Cluster Statistics Collection and Presentation” facility supports both the DECnet and TCP/IP network protocols. The DECnet protocol is available for use by default. To explicitly specify which network protocol to use, define the RDM\$BIND_STT_NETWORK_TRANSPORT logical name to either the value DECNET or TCPIP, respectively. The definition of the logical name must be the same definition on both the local and cluster nodes. You can specify the logical name in the LNM\$FILE_DEV table on the local node, but you must specify the logical name in the LNM\$SYSTEM_TABLE table on all remote nodes. There is no command qualifier to specify the network protocol.

SCREEN HEADER
 REGION CHANGES

With the introduction of clusterwide statistic collection for a database, having the ability to know when clusterwide statistics are being collected is very important. Equally important, however, is knowing whether statistics are being collected on all available cluster nodes on which the database is being accessed.

In releases prior to Oracle Rdb 7.0, it is difficult to determine on how many nodes the database is currently active, as well as the maximum number of nodes on which the database might be active.

The Oracle Rdb 7.0 version RMU Show Statistic utility screen header region includes enhancements to display the current node suffixed with a “(x/y/z)” identifier. The “x” value indicates the number of nodes on which statistics are being actively collected. The “y” value indicates the number of nodes on which the database is being accessed. The “z” value indicates the maximum number of nodes on which the database is accessible. For example, the indicator “(1/3/5)” specifies that statistics are being collected from 1 node, but that the database is currently open on 3 nodes, with a maximum open node count of 5.

The “Node” tag changes to “Cluster” when statistic collection occurs on more than one node. This indicates that clusterwide statistic collection is active.

BINARY FILE
SUPPORT

The OUTPUT qualifier continues to work normally, but when in cluster mode also writes the cluster statistic information to the binary output file.

The “per-screen” statistic collection design minimizes network utilization and improves overall system performance. However, this may affect the writing of “by-area” data to the binary output file when you specify the OUTPUT qualifier. For example, because by-area cluster statistic collection occurs only when displaying a by-area screen, by-area information written to the binary output file may be stale if the current screen is not a by-area statistic screen.

You cannot specify the CLUSTER qualifier when using the INPUT qualifier. Furthermore, the online selection of cluster nodes is not available when replaying binary input files.

When replaying a binary input file, the screen header region also accurately reflects the number of remote nodes representing those statistics in the input file.

PER-PROCESS
STALL MESSAGE
INFORMATION

The cluster collection of per-process stall information automatically detects the binding or unbinding of processes to cluster databases. There is no need to manually refresh the database information on the current node.

When displaying any of the per-process screens that support cluster statistic collection, such as the “Stall Messages” screen, “zooming” on any of the displayed processes will display which node that process is using.

It is possible for you to easily add and remove cluster nodes at runtime. Use the “Cluster Statistics” option located in the Tools menu. You access the Tools menu by using the exclamation point (!) on-screen menu option.

Note that the “Cluster Statistics” option is not available if you are monitoring a specific process using the “Process Monitoring” facility.

The “Cluster Statistics” menu contains up to eleven different options, depending on the current state of cluster statistic collection. Some of these options are not always available. The following sections discuss the various menu options.

manually add cluster node to collection”

This option allows you to manually enter the name of a cluster node for which you wish to start statistic collection. Up to 95 different cluster nodes can be added using this option.

This option is useful for entering a comma-separated list of cluster nodes to be connected.

When you specify a list of cluster nodes, the termination of the network will occur upon encountering any error condition. The loss of connection to one node does not affect the connection to the other connected node. The facility will notify you automatically upon encountering an error condition.



Note that you cannot explicitly add the current node to the cluster statistic collection set because the collection of statistics on the current node is implicit.

select cluster node for collection”

This option displays an alphabetical menu of all available cluster nodes in the cluster that are not currently members of the collection set. This menu allows you to easily select the cluster node for which you wish to start statistic collection. You can add up to 95 different cluster nodes using this option.

remove cluster node from collection”

This option displays a menu of cluster nodes for which the “Cluster Statistic Collection and Presentation” facility is currently active. Selecting one of these active nodes will stop cluster statistic collection from that node. This option is only available when cluster statistics are being actively collected.



Note that you cannot remove the current node from the cluster statistic collection set. Also, removing the last active node causes the disabling of the cluster statistic collection facility.

Display active collection nodes” This option displays a menu of cluster nodes that are currently members of the cluster statistic collection set. This option is available for your informational purposes. This option always displays the current node first, in case you forget which node you are using.

Each line of the display lists the node name, the date when the open of the database occurred on that node, and whether cluster statistic collection is active or inactive. A cluster node becomes inactive when a network transmission error occurs.

Display available cluster nodes” This option displays an alphabetical list of all valid cluster nodes in your cluster. This option is available for informational purposes.

Reconnect to inactive cluster nodes”

This option is available when there are one or more inactive cluster nodes in your cluster statistic collection set. This option is available to provide you the means to manually attempt to reconnect to the cluster node; if unsuccessful, the removal of the cluster node from cluster statistic collection set is automatic.

Add open database nodes to collection”

This option lets you automatically add to the statistic collection set all cluster nodes on which the database is currently open. You use this option primarily when automatic node detection is disabled.

Display inactive collection nodes” This option displays a list of all nodes on which cluster statistic collection is inactive. This screen is available for informational purposes.

Enable / Disable automatic reconnect

This option toggles between enabling and disabling automatic reconnection of inactive cluster nodes.

Enable / Disable automatic node-detection”

This option toggles between enabling and disabling automatic new cluster node detection.

New node detection allows the facility to automatically notify you when new nodes have joined the cluster. You can specify whether or not these new nodes should join the collection set.

able active/passive node-detection”

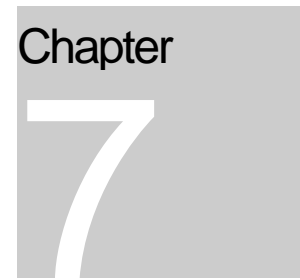
This option toggles between using an active or passive new node detection mechanism. The active detection mechanism requires the RMU Show Statistic utility to read the database rootfile (.RDB) at certain intervals in order to detect new cluster nodes. The passive detection mechanism relies on other active users to passively perform this detection.

For the best performance and least impact on the runtime system, you should use the passive detection mechanism. This is the default behaviour.

Adding a new node or deleting an existing node from the cluster statistic collection automatically resets all of the statistic information so that you have a consistent point-in-time view of the statistic information. If desired, you may unreset the statistics using the “Unreset” on-screen menu option.

S U M M A R Y

The RMU Show Statistic utility “Cluster Statistics Collection and Presentation” facility is a very powerful tool. It provides for the efficient collection of statistical and analytical information from the various database nodes of the cluster. It also provides the ability to drill-down to specific cluster nodes to further analyze performance problems.



Tools

Chapter 7 - Tools

INTRODUCTION

The RMU Show Statistic utility provides a set of tools to create and maintain a “notepad” and to invoke several predefined and user-defined commands. The commands can be invoked while executing the RMU Show Statistic utility.

You can invoke the Tools menu by entering the exclamation point (!) character on any screen. However, you cannot invoke the Tools menu while another named menu is displayed.

The following is an example of the Tools menu:

```
+----- Select Tool -----+
|
| A. Goto screen "by-name"
| B. Goto screen          [->
| C. Notepad (EDT)
| D. Edit a file (EDT)
| E. Send/Receive Mail
| F. Invoke command
| G. Wake up process
| H. Terminate image
| I. Terminate process
| J. Set process priority
| K. Switch database     [->
| L. Start screen cycling
| M. Start stall logging
| N. Start DBKEY logging
| O. Disable auto-refresh
| P. Start ALS process
| Q. <<more>>
|
+-----+
```

TOOLS MENU

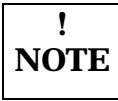
OPTIONS

The following list identifies the options supported by the Tools menu. Note that, depending on database attributes and previously selected tools options, not every option described below will always be available.

- Save current configuration This option exports the current RMU Show Statistic configuration variable definitions into a new configuration file; refer to the Chapter 5 for more information on this option.
- Import configuration settings This option imports a new configuration file at runtime; refer to the Chapter 5 for more information on this option.
- Goto screen “by-name” This option allows you to enter the case-insensitive name of the screen you wish to go to, or any portion of the screen name.
- Note that certain screen names are not unique unless you have entered a sizable portion of the screen name. If you enter a response that is not unique, the first matching screen will be selected. Note that the selections are not necessarily in alphabetical order. If you enter a response that does not match any available screen, the RMU Show Statistic utility displays an error and you will be left on the current screen.
- Goto screen [→ This option displays a multi-page menu of all available screen names, in alphabetical order. The selectable names of the screens depend on the screens available to the RMU Show Statistic utility. For example, the global buffer screen names are not selectable unless they are actually available.
- Notepad (*editor_name*) This option allows you to invoke the specified editor to edit the current notepad.
- Edit a file (*editor_name*) This option allows you to invoke the specified editor to edit a file of your choice.
- Send/Receive mail Invokes the operating system mail utility, allowing you to send and/or receive mail; this is an operating system specific option.
- Invoke command Invokes the operating system command of your choice.
- Wake up process Issues a “wake up” call to a hibernating database process; this option is only available on OpenVMS systems.
- Terminate image This option allows you to terminate an image without terminating the process, if you have sufficient privileges.
- Terminate process This option allows you to terminate a specific database process, if you have sufficient privileges.
- Set process priority This option allows you to alter the operating system priority of a database process, if you have sufficient privileges.
- Switch database [→ This option allows you to switch to another database; the utility will unbind from the current database first.
- You are presented with a menu of all databases that have been opened in the current session, as well as an option to open a new database.

If you choose the “New Database” option, you are prompted for the name of the new database you want to open. You can use command recall to display previous terminal inputs. You can use Ctrl/Z on OpenVMS to cancel the request.

When you enter a new database name, the utility unbinds from the current database and binds to the new database. If the new database cannot be accessed for any reason, the previously accessed database is re-bound again.



Since you actually unbind from the previous database, you will lose any statistic fields that have been put on to the “Custom Statistics” screen.

Also, there are implications when switching databases while collecting statistic information from multiple nodes of the cluster. For more information on these implications, as well as other restrictions, please refer to the “Switching Databases” section in Chapter 6.

The ability to switch to a new database is not available if you are replaying a binary input file, using the INPUT qualifier, or recording a binary output file, using the OUTPUT qualifier.

Stop screen cycling Stops the RMU Show Statistic utility from cycling through the screens in the current menu.

Start screen cycling Starts the RMU Show Statistic utility cycling automatically through the screens in the current menu.

When the RMU Show Statistic utility is cycling through screens, you can change screen modes or select a differ submenu; cycling through the screens associated with your choice will continue at whichever menu level is currently selected.

When you select the “Start screen cycling” option, you are prompted to enter a time interval (in seconds).The specified value for the time interval must be greater than or equal to the value specified for the TIME qualifier. In addition, if you manually change the refresh rate (using the “Set_rate” on-screen menu option) to a value that is greater than the value you specify for the time interval, the cycling is performed at the interval you specify with the “Set_rate” on-screen menu option.

Stop stall logging Stops logging process stall messages to the log file. The log file is closed and may be viewed.

Verbose stall logging This option indicates that all active stall messages should be logged to the stall message log file.

Brief stall logging This option indicates that only new stall messages should be logged to the stall message log file.

Start stall logging Starts logging process stall messages to a log file of your choice.

You are prompted to enter a file name for the stall log. Writing stall messages to a log file can be useful when you notice a great number of stall messages being generated, but do not have the resources to immediately investigate and resolve the problem. The log file generated can be reviewed later so that the problems can be traced and resolved.

The stall messages are written to the file in a format similar to the Stall Messages screen. Stall messages are written to the file at the same rate as the screen refresh rate. Specifying a large refresh rate minimizes the size of the file, but results in a large number of missed stall messages. Specifying a small refresh rate produces a large log file, but contains more of the stall messages generated.

You do not need to display the Stall Messages screen to record the stall messages to the log file. The stall log is maintained regardless of which screen, if any, is displayed.

- Stop lock timeout logging Stops logging lock timeout messages to the log file.
- Start lock timeout logging Starts logging lock timeout messages to a log file of your choice.
- Stop lock deadlock logging Stops logging lock deadlock messages to the log file.
- Start lock deadlock logging Starts logging lock deadlock messages to a log file of your choice.
- Stop DBKEY logging Stops logging process DBKEY access messages to the log file.
- Start DBKEY logging Starts logging process DBKEY access messages to a log file of your choice.

You are prompted to enter a file name where all the accessed DBKEYs are logged.

The DBKEY access information is written at the current screen refresh rate. Using a larger refresh rate minimizes the size of the file but results in a large number of missed DBKEY messages. Using a smaller refresh rate produces a large log file, but contains a much finer granularity of DBKEY messages.

Note that you do not need to display the “DBKEY Information” screen to record the DBKEY messages to the DBKEY log file. The DBKEY log is maintained regardless of which screen, if any, is displayed.

- Disable automatic refresh Disables the automatic refreshing of the screen status region following a screen status update; you will have to manually issue the Ctrl/W keystroke to refresh the screen.
- Enable automatic refresh Enables the automatic refreshing of the screen status region following a screen status update; you do not have to manually issue the Ctrl/W keystroke to refresh the screen.
- Selective alarm bell This option indicates that the alarm bell is to be sounded only when a new alarm message is generated
- Continuous alarm bell This option indicates that the alarm bell is to be sounded if any alarm message is generated.

Alarm classes	Identifies the alarm classes that should be used when generating alarm messages.
Stop ALS process	Stops the AIJ Log Server (ALS) process; this is the same as issuing the RMU Server After_Journal Stop command.
Start ALS process	Starts the AIJ Log Server (ALS) process; this is the same as issuing the RMU Server After_Journal Start command.
Invoke ABS process	Invokes the AIJ Backup Server (ABS) process for a specific after-image journal; there is no corresponding RMU Server command
Resume ABS process	Resumes allowing AIJ Backup Server (ABS) processes to be invoked on after-image journal switch-over operations; this is the same as issuing the RMU Server Backup Resume command.
Suspend ABS process	Suspends AIJ Backup Server (ABS) processes from being invoked on after-image journal switch-over operations; this is the same as issuing the RMU Server Backup Suspend command.
Wake up RCS process	Issues a “wake up” to the Row Cache Server (RCS) process.
Cluster statistics [→]	This option displays a submenu of cluster statistics processing options.
Refresh cluster information	Refreshes the clusterwide information on the current node.
Logical area menu filter	Identifies the filter, if any, to be used for the logical area menu; typically used for databases with an extremely large number of logical areas.
Display application logical areas	This option indicates that only application-defined logical areas (i.e. not system-defined) should be displayed in the logical area menu.
Display all logical areas	This option indicates that all logical areas, including system-defined logical areas, should be displayed in the logical area menu.
Display page information	This option allows you to display the contents of a storage area page. This option is not available when replaying a binary input file.
Disable “exit” confirmation	This option indicates that you do not wish to confirm the “exit” on-screen menu option. This option affects the CONFIRM_EXIT configuration variable.
Enable “exit” confirmation	This option indicates that you wish to confirm the “exit” on-screen menu option. This option affects the CONFIRM_EXIT configuration variable.
Set user-prompt timeout	This option allows you to enter a new user-prompt timeout interval, or disable the timeout altogether. This option affects the PROMPT_TIMEOUT configuration variable.

Start automatic screen capture¹⁵ This option allows you to automatically capture images of all screens, at a specified interval. The facility is similar to manually using the “Options” on-screen menu option every so often.

Only the numeric display format of screens is captured.

You will be requested to enter the interval between screen capture operations, expressed in seconds. The minimum interval is 30 seconds.

It takes approximately 5 to 10 seconds to capture all available screens. You will be notified when the screens are being captured by the message “*** Writing Report ***” being displayed on the status region of the current screen.

In order to guarantee consistent statistic information, statistic information updates are temporarily “paused” while the screen capture operation is occurring. Note that this “pause” also affects writing to the binary output file, as well as any log files being recorded.

The “Automatic Screen Capture” facility can also be invoked using the REPORT_INTERVAL configuration variable specifying the number of seconds.

There is no corresponding command qualifier for this facility.

This option is not available if the NOINTERACTIVE qualifier is specified.

The “Automatic Screen Capture” facility works with binary files and can be used if the INPUT qualifier is specified.

The “Automatic Screen Capture” facility is integrated with the cluster statistic collection facility. If cluster statistic collection is enabled, all supported screens will provide cluster information.

Process monitoring [->]¹⁶ This option displays a submenu of RMU Show Statistic utility process monitoring options; the submenu remains displayed until you enter Ctrl/Z on OpenVMS.

The “Process Monitoring” facility submenu options are the following:

Option	Description
Attach to activated processes	This option enables the display of statistics for any processes that are globally collecting statistics information. These processes are identified by the “A” suffix. This option is automatically performed when displaying any

¹⁵ Available in Oracle Rdb 7.0.2.0.

¹⁶ Available in Oracle Rdb 8.0.

<u>Option</u>	<u>Description</u>
	<p>of the process overview information screens.</p> <p>This option is really only useful when you have not visited the process overview screens and you wish to monitor a specific process.</p>
Detach from activated processes	<p>This option disables the display of statistics for processes that are globally collecting statistics information.</p> <p>This option is not available when displaying any of the process overview information screens, or if no processes are being monitored.</p>
Activate process statistics collection	<p>This option notifies a selected process to activate its global statistic collection. This option can only be used on processes NOT displaying the “A” suffix.</p> <p>This option is automatically performed if you opened the database using the STATISTIC=PROCESS_GLOBAL qualifier to the RMU Open command. The RMU Show Statistic utility will wait no more than 30 seconds for the selected process to activate the process global statistic collection. If the process is in the midst of a long operation, the activation request may expire and you will have to re-issue the activation request at a later time.</p> <p>This option will automatically attach the RMU Show Statistic utility to the process, if the activation is successful. This option cannot be used on “server” processes, such as the AIJ Log Server (ALS) and Record Cache Server (RCS). Server processes must be manually activated by setting the RDM\$BIND_XXX_GLOBAL_STATISTICS system logical name to the value “1”, where the “xxx” designation is ABS, ALS, LCS or LRS.</p> <p>Note that the DBR does not collect global statistic information, because it is a short-lived process.</p>
De-activate process statistics collection	<p>This option notifies a selected process to de-activate its global statistic collection. This option can only be used on processes displaying the “A” or “G” suffixes.</p> <p>This option is not available if you opened the database using the STATISTIC=PROCESS_GLOBAL qualifier to the RMU Open command.</p> <p>NOTE: If multiple RMU Show Statistic utilities are moni-</p>

<u>Option</u>	<u>Description</u>
	<p>toring the same process, this option will cause the process to de-activate the global collection, which other RMU Show Statistic utility viewers will view as cessation of activity by that user. No notification to other RMU Show Statistic utility viewers is provided, although they will be notified if the specific process being monitored was de-activated.</p> <p>This option cannot be used on “server” processes, such as the AIJ Log Server (ALS) and Record Cache Server (RCS).</p>
<p>Select attached process to monitor</p>	<p>This option allows you to select a single process to monitor. The selected process must have been previously attached, either implicitly by visiting any of the process overview screens, or explicitly by using “Attach to activated processes” option above.</p> <p>All of the RMU Show Statistic utility screens display information for the selected process only; the “Node:” section of the screen header region is replaced with the PID of the selected process. The statistic information displayed is automatically reset when this option is selected. However, you can use the “Unreset” on-screen menu option to restore the full statistic information.</p> <p>If you are recording a binary output file, a new file will be automatically created. The new binary output file will be for the selected process only.</p>
<p>Cancel process monitoring</p>	<p>This option reverts the statistic information display back to its normal state, where the screens reflect all processes on the database.</p> <p>If you are recording a binary output file, a new file will be automatically created. The new binary output file will be for all database processes. The statistic information displayed is automatically reset when this option is selected. However, you can use the “Unreset” on-screen menu option to restore the full statistic information. This option is automatic if the process being monitored unbinds from the database, or has its global statistic collection deactivated by another RMU Show Statistic utility user.</p>

Stop automatic screen capture¹⁷ This option stops the automatic capture of screen information.

HOW STATISTIC options [-> This option displays a submenu of RMU Show Statistic utility processing options; the submenu remains displayed until you enter Ctrl/Z on OpenVMS.

The processing options are the following:

<u>Option</u>	<u>Description</u>
Use EDT Editor	This option enables the EDT editor, used for the “Notepad” facility and for editing ordinary files.
Use LSE Editor	This option enables the LSE editor, used for the “Notepad” facility and for editing ordinary files.
Use TPU Editor	This option enables the TPU editor, used for the “Notepad” facility and for editing ordinary files.
Change notepad	This option allows you to specify a new file name for the notepad. Typically, you want to change the notepad name to RMU.SCR or STATISTICS.RPT to edit screen captures. The default notepad name is STATS.NOTEPAD.
Enable Time_plot baseline	This option enables the “Time_plot” screen format displays to always display the current time plot, even on the baseline of the chart.
Disable Time_plot baseline	This option disables the “Time_plot screen format displays from displaying statistic inactivity on the baseline of the chart.

NOTEPAD

INFORMATION

The notepad is a file that you can use to maintain notes about a session. The name of the notepad file can be specified using the “SHOW STATISTIC options [->” option of the Tools menu.

The default notepad file name is STATS.NOTEPAD.

If the notepad file does not exist, it is automatically created containing four lines of information before the designated text editor is invoked. The information identifies the product version, the database name, the notepad file name, as well as the date and time the notepad file was originally created.

¹⁷ Available in Oracle Rdb 7.0.2.0.

You might find it useful to define the notepad file name to be one of the other RMU Show Statistic utility file names, such as RMU.SCR, which is the name of the file used for writing screens. You can now create an image of a particular screen using the Write menu option, and then document it using the “Notepad” facility.

NOTEPAD EDITORS

On OpenVMS, you can invoke one of three text editors: EDT, LSE, or TPU. You can change the editor by using the Options submenu (described later) or by defining the RMU\$EDIT logical name. The value for the RMU\$EDIT logical name must be EDT, LSE, or TPU. The default editor is EDT.

If you have an EDTINI.EDT file, it is read when the text editor is invoked.

SENDING AND/OR
RECEIVING MAIL

On OpenVMS, you can send or receive email without having to exit the RMU Show Statistic utility. The “Send/Receive mail” option of the Tools menu invokes the operating system mail utility.

If you specified the /BROADCAST qualifier and have your terminal enabled to receive email notification, the RMU Show Statistic utility will automatically notify you when new mail arrives.

In order to receive new email notification messages, the following two OpenVMS DCL commands must be issued:

```
$SET TERM /BROADCAST
$SET BROADCAST=MAIL
```

On OpenVMS, you can also send and receive email by entering the at sign (@) character on any screen.¹⁸

INVOKING A
COMMAND

The “Invoke command” executes a user-specified operating system command. On OpenVMS, the DCL (\$) prompt is then displayed at the bottom of the current screen and you can enter any DCL command.

On OpenVMS, you can also invoke the DCL command facility by entering the dollar sign (\$) on any screen. The (\$) prompt is then displayed on the bottom of the current screen, and you can enter any DCL command.

You should be careful when invoking any utility (such as SQL) that operates against the database. Any command that requires exclusive database access will hang indefinitely.

¹⁸ Available starting with Oracle Rdb 7.0.2.0.

WAKE UP PROCESS

This option wakes up a hibernating process, which is typically either the AII Log Server (ALS) or the Row Cache Server (RCS) processes. However, when using the ALS server, application processes will also hibernate if they are attached to only one database.

This option should only be necessary following catastrophic database failure where the Database Recovery (DBR) process was unable to wake up the process. This option has no effect on a process that is not hibernating; however, take care not to overuse this option.

This option is limited to operations performed on specific database processes; you cannot use this option on arbitrary processes on the system.

TERMINATE IMAGE

This option terminates a database image with the following exception:

“%RDMS-F-IMGABORTED, image aborted at privileged user request”.

Note that the process itself is not terminated, only the image is terminated.

This is the preferred operation as this operation does not cause the database to be frozen during process recovery.

This option is limited to operations performed on specific database processes; you cannot use this option on arbitrary processes on the system.

TERMINATE
PROCESS

This option terminates a database process without warning.

This operation causes the database to be frozen while the Database Recovery (DBR) process recovers the terminated process. This option is limited to operations performed on specific database processes; you cannot use this option on arbitrary processes on the system.

SET PROCESS
PRIORITY

This option allows you to change the priority for the specified process on OpenVMS. You need the ALTPRI (alter priority) privilege to set the priority higher than the base priority of the specified process.

User-Defined Events

Chapter 8 - User-Defined Events

INTRODUCTION

Most database products provide tools for the purposes of analyzing the performance characteristics of the database and the application. These tools often require constant manual attention, or extensive post-processing of recorded data, to detect critical events that might be detrimental to the smooth operations of the database.

In some cases, the DBA is not always available to constantly monitor the utility output, especially for “24*7” production databases. Post-processing of recorded data is certainly not timely enough to prevent or resolve potential realtime database downtime situations, particularly with real-world mission-critical requirements. The DBA needs the ability to be “notified” automatically when a time-critical or interesting events occur to the database.

The RMU Show Statistic utility is one of the most powerful and useful tools available to the DBA for the purposes of analyzing performance characteristics of a database. However, the ability to analyze performance problems is just one aspect of role of a performance analysis tool. The DBA needs to be able to detect problems in a timely manner, analyze the problem, and immediately institute corrective actions, also in a timely manner.

ORACLE RDB7

The RMU Show Statistic utility has been dramatically enhanced for Oracle Rdb 7.0 to include many new features that provide DBAs with the information necessary to detect, analyze and correct performance problems in the database. Even more enhancements have been added to the Oracle Rdb 7.0.1.0 release, including the “Configuration File” facility, which provides an extremely powerful and easy-to-use “User-Defined Event” facility.

Through the use of the RMU Show Statistic utility configuration file, you are able to persistently define special “events” that specify the action to be performed when something of interest occurs to the runtime database. An event is an identification of a

particular statistic value upon which the RMU Show Statistic utility is to perform some user-defined action. In other words, an event is signaled when a statistic's value exceeds a user-defined set of thresholds.

Through the use of events, the DBA is able to be automatically notified by a RMU Show Statistic utility server running on behalf of a particular database. An event is defined by specifying a threshold against a specific statistic, and optionally specifying the attributes the event is to have.

Event Definition Syntax

The RMU Show Statistic utility configuration file is a human-readable file, created using the editor of your choice. The configuration file typically resides in the database directory, although it can reside anywhere that you desire.

Each entry in the configuration file uses the general format `variable=value`; The equal sign (=) separating the variable and value is required. Also note that each definition is terminated with a semicolon character (;).

User-defined events are specified using the `EVENT_DESCRIPTION` variable. Events themselves are not named; rather, they are defined on behalf of a specific statistic for a given threshold. The `EVENT_DESCRIPTION` variable's value is a free-format description of the user-specified event. The event definition consists of three required position-dependent components and an optional component; the following example describes the general format:

```
EVENT_DESCRIPTION="operation \  

  statistic_name \  

  threshold_name \  

  [ attribute_list ]";
```

" OPERATION "
 CLAUSE

The event "operation" clause identifies the action to be performed for the `EVENT_DESCRIPTION` operation. The keyword `ENABLE` is available for use to enable a new event or change an existing event definition. The keyword `DISABLE` is available for use to disable a previously defined event; this keyword is typically used when importing a new configuration file.

Even though an event may be enabled, it may not be active. In order for an event to be active, you must also specify either a program to be invoked or one or more operator classes to be notified. This requirement is discussed later in this document.

During runtime, events can only be disabled by the RMU Show Statistic utility, or by importing a new configuration file that explicitly disables an event.

“ STATISTIC NAME ”
 C L A U S E

The event “statistic name” clause identifies the particular valid statistic field for which the event will be enabled or disabled. Note that some statistic names are only valid when certain database attributes are enabled, such as global buffers or record caching. The names of a particular statistic field can be found on the individual screen of interest. When statistic field names contain multiple words, such as “process attaches”, it is required that the statistic name be either single or double quoted; failure to quote the statistic name may result in a syntax error.

Certain statistic fields have leading blanks. This whitespace is considered part of the statistic name, and is necessary to identify a unique statistic. However, the use of leading blanks is often difficult to discern in the configuration file. Therefore, the underscore character (_) or dash character (-) can be used in place of spaces in statistic names that have leading spaces. For example, the statistic field name “ file extend” can also be specified as “ _ _ file_extend” or “ - - file-extend”. This is useful for improving the readability of difficult statistic field names.

Aggregated Statistic Names

Most general events are defined using the summary statistics screens. However, it is sometimes necessary to define an event on a specific table or index, or even a particular partition of a table. This type of drill-down event is specified using the AREA attribute (described later) to specify the name of a particular storage area. When this clause is specified, the statistic field selected must be from the “IO Statistics (by file)” or “Locking Statistics (by file)” screens. The identified statistic name can also be qualified with the LAREA attribute (described later) to specify the name of a particular logical area, such as a table, b-tree index, hash index, or blob. When this clause is specified, the statistic field selected must be from the “Logical Area” screens. Furthermore, if the selected logical area is partitioned across multiple storage areas, the AREA clause can also be used to identify a specific partition to define the event against.

The following table explains the semantics of specifying the AREA and LAREA clauses to qualify a statistic field name:

<u>AREA</u>	<u>LAREA</u>	<u>Description</u>
No	No	General Statistic Field
Yes	No	Storage Area Statistic Field
No	Yes	Logical Area Statistic Field, all partitions
Yes	Yes	Logical Area Statistic Field, single partition

The AREA and LAREA clauses are attributes and, as such, must follow the “Threshold Name” component of the event definition.

Unnamed Statistics Sometime, it is necessary to define an event for an unnamed statistic, such as the transaction duration information. This is achieved using the special “CELL number” statistic name syntax¹⁹. By specifying a cell number instead of a statistic name, you are able to specify events on any statistic, even if the statistic name is not available.

You can also use the ANNOTATE attribute to specify a statistic name for the selected cell number. If you do not use the ANNOTATE attribute, the statistic name will be the specified text “CELL number”.

The cell numbers are documented in Appendix A, as well as in the text file

SYSSLIBRARY:RMU\$SHOW STATISTICS.CDO

If you have installed multiple versions of Oracle Rdb on your system, the text file is named

SYSSLIBRARY:RMU\$SHOW STATISTICS w .CDO

where w is the Oracle Rdb software version (i.e. “70”).

“THRESHOLD
 NAME” CLAUSE

The event “threshold name” clause identifies the particular event threshold for which the specified statistic will be enabled or disabled. The thresholds are essentially the columns in the numeric version of the statistics screens.

Up to eight different thresholds can be specified for a particular statistic field, although each individual event name must be specified in its own EVENT_DESCRIPTION variable definition. The “threshold name” clauses are the following:

- MAX_RATE The maximum current occurrence-per-second rate collected. This threshold only increases as each event is signaled.
- MAX_CUR_TOTAL The maximum total value collected since the database was opened. This threshold only increases as each event is signaled.
- MIN_CUR_RATE The lowest rate currently being sustained. This threshold remains constant.
- MAX_CUR_RATE The highest rate currently being sustained. This threshold remains constant.
- MIN_AVG_RATE The lowest average rate. This threshold only decreases as each event is signaled.
- MAX_AVG_RATE The highest average rate. This threshold only increases as each event is signaled.
- MIN_PER_TX The lowest per-transaction rate. This threshold only decreases as each event is signaled.

¹⁹ Available in Oracle Rdb 7.0.2.0.

MAX_PER_TX The highest per-transaction rate. This threshold only increases as each event is signaled.

“ATTRIBUTE LIST”
 CLAUSE

The optional event “attribute list” clause provides additional characteristics for enabled event thresholds; in general, these attributes are ignored when disabling an event. Any or all of the event attributes can be specified for each event name within the same EVENT_DESCRIPTION variable definition. The attribute list clauses are the following:

AREA *storage_area_name* Defines the name of a particular storage area. When this clause is specified, the statistic field selected must be from the “IO Statistics (by file)” or “Locking Statistics (by file)” screens, unless the LAREA clause is also specified. This clause is not ignored when disabling the event.

LAREA *logical_area_name* Defines the name of a particular logical area, such as a table, b-tree index, hash index, or blob. When this clause is specified, the statistic field selected must be from the “Logical Area” screens. This clause is not ignored when disabling the event.

This clause cannot be used if the NOLOGICAL_AREA qualifier is specified.

INITIAL *value* Defines the initial value of the current event threshold; the default value is “0” for MAX_XXX thresholds and “very big number” for MIN_XXX thresholds. The default value guarantees that at least one event will be signaled, thereby initializing the new current threshold value.

EVERY *value* Defines the value by which the initial threshold will be incremented or decremented when an event is signaled. If this value is “0”, the default value, for any event except the MIN_CUR_RATE and MAX_CUR_RATE events, then the event will be signaled just once.

LIMIT *value* Defines the maximum number of times the event may be signaled. If the value is “0”, the default value, events may be signaled indefinitely, if the EVERY clause is specified with a non-zero value.

SKIP *value* Defines the number of event notifications to ignore before performing an actual notification. This clause is extremely useful for the MIN_CUR_RATE and MAX_CUR_RATE events, as the thresholds for these events are not reset upon being signaled. The default value of “0” ensures that all events are notified.

ANNOTATE *stat_name*^{e0} Defines the name of a statistic specified using the “CELL *number*” syntax. The maximum length of the specified statistic name is 18 characters. This attribute can be used even if a statistic name is specified instead of its cell number.

^{e0} Available in Oracle Rdb 7.0.2.0.

- NOTIFY *oper_class_list* Defines the quoted comma-separated list of operators to be notified for all events defined on the specified statistic. Valid operator keywords are CENTRAL, DISKS, CLUSTER, SECURITY and OPER1 through OPER12.
- INVOKE *program_name* Defines the user-supplied program to be invoked for all events defined on the specified statistic. On OpenVMS, the program name is specified as a DCL process global symbol known to the RMU Show Statistic utility.

DESCRIPTION
 READABILITY

Really long entries in the configuration file lines can be continued on the next line by terminating the line with a back-slash (\). The continuation character as the last character of a line is available for use to indicate that the configuration entry is continued on the next line exactly as if it were entered as a single line. Lines can be continued practically indefinitely, up to 2048 characters, even within quoted values. The following example demonstrates how to define a multi-line event description:

```
EVENT_DESCRIPTION="ENABLE 'pages checked' \  

  MAX_CUR_TOTAL \  

  INITIAL 7 \  

  EVERY 11 \  

  LIMIT 100 \  

  INVOKE DB_ALERT";
```

The continuation character is not limited to the EVENT_DESCRIPTION variable; it can be used for any configuration variable.

Comments can be embedded in continued lines if they start at the beginning of the next line. The following example demonstrates two event descriptions containing embedded comments. The comment in the second event description takes precedence over the line continuation character:

```
EVENT_DESCRIPTION="ENABLE ' (Asynch. reads)' \  

  MAX_CUR_TOTAL \  

  AREA EMPIDS_OVER \  

  ! this will work as expected  

  INITIAL 6 EVERY 10 LIMIT 100 \  

  INVOKE DB_ALERT";
```

```
EVENT_DESCRIPTION="ENABLE ' (Asynch. reads)' \  

  MAX_CUR_TOTAL ! this will NOT work as expected \  

  AREA EMPIDS_OVER \  

  INITIAL 6 EVERY 10 LIMIT 100 \  

  INVOKE DB_ALERT";
```

Event Semantics

In order for an event to be active, you must specify either one or both of the NOTIFY or INVOKE attribute clauses. When using the INVOKE attribute clause, the program must be specified by defining a process-global symbol pointing to the DCL command procedure or image to be invoked. The INVOKE program and NOTIFY operator classes apply to all events defined for the statistic field. Therefore, these clauses only need to be defined once per statistic field, no matter how many events thresholds are defined for that statistic. Specifying multiple programs or operator classes results in only the last specified attribute to be used.

Once an event has been signaled, it will only be re-signaled if the EVERY attribute clause was specified with a non-zero value. The current threshold value, originally initialized to the INITIAL value, will be advanced for MAX_XXX thresholds and declined for MIN_XXX thresholds. The exception to this rule is the “current rate” thresholds MIN_CUR_RATE and MAX_CUR_RATE, which are never advanced nor declined. The MIN_XXX thresholds disable themselves once the INITIAL value reaches “0”, while the MAX_XXX thresholds never disable themselves.

Once an event has been disabled, it can only be re-enabled by importing a new configuration file or manually using the “Statistics Event Information” screen “Re-enable all disabled events” configuration submenu option; individual events cannot be re-enabled online.

The user-defined events are analyzed by the RMU Show Statistic utility at the specified screen refresh rate. The default screen refresh rate of three seconds is ideal for most databases, but using a one-second refresh rate will produce a finer granularity event signaling mechanism. Multiple events defined for the same statistic field may cause the specified program to be invoked multiple times, once for each affected event.

As the RMU Show Statistic utility identifies a statistic field whose current value or average value is changing, it examines any defined event thresholds established for that statistic field. This manner of examination minimizes the impact of event analysis, since the analysis is performed as part of the normal statistics collection process.

When the utility determines that a specified event threshold has been exceeded, an event is signaled. The signaling of the event means that any specified programs will be invoked and any specified operators will be notified. The event notification occurs immediately.

If you defined a program that will be invoked when an event is signaled, the program will be invoked with eight parameters. Some of the parameters contain multiple words; be sure to quote them if the parameters are passed to other utilities. The parameters passed to invoked programs are the following:

<u>Parameter</u>	<u>Description</u>
P1	the “date and time” the event occurred. This parameter contains embedded blanks.
P2	the statistic field name. This parameter may contain embedded blanks.
P3	the event name.
P4	the current event numeric value, expressed to the nearest tenth.
P5	the word “above” or “below.”
P6	the current event threshold value.
P7	the event occurrence count.
P8	the optional physical area and/or logical area name for the statistic field.

The P8 parameter is either null (“”) or contains the name of the affected storage area and/or logical area.

The following example contains log file sample output where an event for a partitioned logical area was signaled. Note that when both the storage area and logical area names are specified, they are separated by a period (.).

```
pages checked MAX_CUR_TOTAL 6.0 above 4.0 count is 1 area is
EMPIDS_MID.EMPLOYEES
```

```
pages checked MAX_CUR_TOTAL 32820.0 above 5.0 count is 1 area is
EMPIDS_OVER.EMPLOYEES
```

RUN-TIME EVENT
 STATUS
 INFORMATION

The current runtime status of the user-defined events can be examined using the new “Statistics Event Information” screen, located in the “Database Parameter Info” submenu. Note that you do *not* have to be viewing this screen to signal events. Note that the physical area and logical area identifiers are only displayed in “Full” mode.

Real-Life User-Defined Event Example

Nothing demonstrates a feature better than a real-life example explained in step-by-step detail.

For the purposes of this example, suppose the DBA wants to be sent email whenever a database freeze occurs. A database freeze occurs when an application process on the database prematurely terminates (i.e. “dies”). Such an event results in all application activity being temporarily suspended until the recovery operation for the terminated process has been completed. This is a very significant and serious runtime event that should be immediately detected.

Using events, notifying the DBA when a process terminates prematurely is very easy to accomplish. The following steps describe how this can be achieved using the RMU Show Statistic “User-Defined Events” feature:

IDENTIFY THE
OPERATION

You are going to define a new event, so you need to specify the ENABLE operation keyword.

IDENTIFY THE
STATISTIC NAME

A database freeze is represented by the “process failures” statistic on the “Recovery Statistics” screen, which is located in the “Journaling Information” submenu. This statistic is available even if you are not using after-image journaling.

IDENTIFY THE
THRESHOLD NAME

The current number of processes that have failed, thereby causing a database freeze, is represented by the MAX_CUR_TOTAL threshold.

IDENTIFY THE
EVENT
ATTRIBUTES

This is probably the hardest part of defining an event. You want to be alerted to *any* process failure, so you have to set the INITIAL attribute to “0”. Since you want to be notified on each and every process failure, then you have to set the EVERY attribute to “1” and the LIMIT attribute to “0”.

DEFINE HOW YOU
WILL BE ALERTED

Since you want to be sent mail, use the INVOKE clause. Invoking a program on OpenVMS requires that you define a “DCL process-global symbol” to identify the actual DCL script, as the following example shows:

```
DBR_LOGGER ::= @SYSSYSTEM:DBR_LOGGER.COM
```

WRITE THE
PROGRAM

Since you want to be sent mail with a nice description of what event was actually signaled, use the simple DCL script in the following example:

```
$ create /nolog sys$scratch:dbr_logger.tmp
EOD
$ open /write dbr_logger - sys$scratch:dbr_logger.tmp
$ write dbr_logger "'p1' 'p2' 'p3' 'p4' ", -
" 'p5' 'p6' (count is 'p7') area is 'p8' "
$ close dbr_logger
$ mail sys$scratch:dbr_logger.tmp -
RDB_DBA_USER /subject="DBR notification"
$ delete /nolog sys$scratch:dbr_logger.tmp;*
$ exit
```

CREATE THE
CONFIGURATION
FILE ENTRY

The following example contains the final event description as you would enter it in the configuration file:

```
EVENT_DESCRIPTION="ENABLE 'process failures' \
MAX_CUR_TOTAL \
INITIAL 0 EVERY 1 LIMIT 0 \
INVOKE DBR_LOGGER";
```

INVOKE THE RMU
SHOW STATISTIC
UTILITY

Be sure to use the CLUSTER command qualifier if you want to be notified of clusterwide events. The following example demonstrates the command to perform this:

```
$ RMU Show Statistic -
/CONFIG=CONFIG.CFG -
/NOINTERACTIVE /UNTIL=HH:MM:SS -
MF_PERSONNEL
```

SUMMARY

As applications increasingly require 24*7 availability, the rate at which DBAs are expected to react to potential downtime situations also increases. The "User-Defined Events" feature provides the means by which DBAs are able to be automatically alerted when such critical situations arise, thereby allowing timely corrective actions.

Database Dashboard

Chapter 9 - Database Dashboard

INTRODUCTION

One of the hardest problems facing the DBA performing database performance characterization, monitoring and problem analysis is the determination of the actual runtime attributes of the database and the processes attached to the database. On OpenVMS, processes can specify “local” attributes in the form of process or job logical names, and the DBA is unable to determine those settings using tools provided by the operating system. Knowledge of these settings is not only vital to the proper operation of the database, but is also necessary to prevent unauthorized consumption of system resources.

Oracle Rdb 7.0 introduces the “Database Dashboard” feature to the RMU Show Statistic utility, for both the global database and individual database processes. The purpose of the “Database Dashboard” is to display the actual database parameter and attribute settings being used by the processes actively attached to the database and, optionally, allow the dynamic and non-persistent modification of certain database parameters and attributes on a single node, at runtime. The changing of these parameters can also be actively or passively broadcast to all active database processes on the node upon which the changes were made. The net effect of these changes can then be monitored at runtime without having to restart database processes.

For example, when the RDM\$BIND_BUFFER logical name is defined at the process level, there is no method available for the DBA to examine its setting at runtime for a particular attached process. The same is true for any database logical name, as well as most interesting database attributes that affect runtime performance of the database.

The facility is known as the “Database Dashboard” because you can actually drive the database faster or slower, and immediately see the impact or benefit of changing certain database settings higher or lower.

SECURITY

To update a dashboard element of the “Database Dashboard” facility, you need the proper privilege. On OpenVMS, you must have WORLD and BYPASS privilege to

be able to dynamically update any dashboard screen.²¹ The RMU Show Statistic utility must also be invoked using the `OPTION=UPDATE` qualifier.

INVOCATION

The “Database Dashboard” is an integral component of the RMU Show Statistic utility, and is available during online access of the database. The console is not available during replay of binary input files because the console displays realtime database parameter information that is not available in the binary input file.

DASHBOARD
SCREENS DISPLAY
FORMAT

The “Database Dashboard” facility consists of a series of screens containing runtime database information. A particular dashboard screen is comprised of a set of related dashboard elements, such as “I/O Information” or “AIJ Information”. Each dashboard element represents a database attribute, some of which can be modified at runtime. Others are merely displayed for informational purposes, so that the DBA can analyze current database settings.

There are two types of dashboard screens: global and process. The global dashboard screens contain dashboard elements that pertain to all processes attached to the database; that is, “global” information. The process dashboard screens contain dashboard elements for specific database processes attached to the database on that particular node. Typically, the global dashboard elements are the “default” values for the process dashboard elements.

²¹ On Digital UNIX, no updating is currently permitted.

The following is an example of a global dashboard screen:

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 7-FEB-1998 06:56:26.54
Rate: 1.00 Second IO Dashboard Elapsed: 00:00:41.43
Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

Database.....	Current...	Previous..	Lowest....	Highest...	Original..	Chng
Attribute.Name....	Value.....	Value.....	Value.....	Value.....	Value.....	Cnt.
Buffer Count	20	20	20	20	20	0
APF Enabled	1	1	1	1	1	0
APF Depth	5	5	5	5	5	0
DAPF Enabled	1	1	1	1	1	0
DAPF Depth Count	4	4	4	4	4	0
DAPF Start Count	4	4	4	4	4	0
ABW Enabled	1	1	1	1	1	0
ABW Clean BufCount	5	5	5	5	5	0
ABW Batch Max	4	4	4	4	4	0

Config Exit Help Menu Options Set_rate Update Write !

The following is an example of a process dashboard screen:

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 8-FEB-1998 07:00:33.99
Rate: 1.00 Second Per-Process I/O Dashboard Elapsed: 00:01:29.84
Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

Database.....	Current...	Previous..	Lowest....	Highest...	Original..	Chng
Attribute.Name....	Value.....	Value.....	Value.....	Value.....	Value.....	Cnt.
For Process: 38211CB8:1 - RICK7						
Buffer Count	500	500	500	500	500	0
APF Enabled	1	1	1	1	1	0
APF Depth	5	5	5	5	5	0
DAPF Enabled	1	1	1	1	1	0
DAPF Depth Count	4	4	4	4	4	0
DAPF Start Count	4	4	4	4	4	0
ABW Enabled	1	1	1	1	1	0
ABW Clean BufCount	5	5	5	5	5	0
ABW Batch Max	4	4	4	4	4	0
Lock Timeout Intvl	2147483647	2147483647	2147483647	2147483647	2147483647	0
Ready AreaSerially	0	0	0	0	0	0
Snap Quiet Point	1	1	1	1	1	0
Hold Retrvl Locks	0	0	0	0	0	0
Coarse Buf Lockng	0	0	0	0	0	0

Config Exit Help Menu Options Set_rate Update Write !

DATABASE
 DASHBOARD
 SCREEN
 CONTENTS

Each of the “Database Dashboard” screens contain seven columns of information. These columns provide information about the database parameters, including any modifications the DBA may have made to those parameters.

Database Attribute Name	This is the database dashboard attribute.
Current Value	The Current Value column contains the current value for each dashboard element. On global database screens, this is the default value being used by all processes. On the process screens, this is the actual value being used by that particular process.
Previous Value	The Previous Value column contains the previous value for each database attribute. When a database attribute is dynamically updated by the DBA, this column will contain the value before it was modified.
Lowest Value	The Lowest Value column contains the lowest modified value for each database attribute.
Highest Value	The Highest Value column contains the highest modified value for each database attribute.
Original Value	The Original Value column contains the original value for each database attribute. This column remains unchanged regardless of the modifications made to the database attribute.
Chng Cnt	The Chng Cnt column indicates the number of changes to each database attribute. On global database screens, this count is the number of times the particular database attribute was updated by the user. On the per-process screens, this count is the number of times the actual process has updated its dynamic information.

DASHBOARD
 SCREENS

The “Database Dashboard” facility consists of a large number of screen “categories”. Each screen contains commonly-related dashboard items. The available screens are:

IO Dashboard	The “IO Dashboard” screen contains the global database I/O information that can be dynamically modified. Selectable fields include the default buffer count used by database processes, the asynchronous pre-fetch depth setting, the detected asynchronous pre-fetch depth count setting, the detected asynchronous pre-fetch start count, the asynchronous batch write clean buffer count setting and the asynchronous batch write maximum batch size setting.
Locking Dashboard	The “Locking Dashboard” screen contains the global database locking information that can be dynamically modified. Selectable fields include the lock timeout interval and several boolean attributes.

- AIJ Dashboard** The “AIJ Dashboard” screen contains the global database after-image journal information that can be dynamically modified. Selectable fields include the minimum AIJ group commit I/O buffer size, in blocks, the maximum AIJ group commit I/O buffer size, in blocks, the AIJ group commit stall time, in milliseconds and the TSNBLK group commit stall time, in milliseconds.
- Checkpoint Dashboard** The “Checkpoint Dashboard” screen contains the global database checkpoint information that can be dynamically modified. Selectable fields include the various checkpoint thresholds.
- Hot Standby Dashboard** The “Hot Standby Dashboard” screen contains the global database “Hot Standby” feature attributes that can be dynamically modified. Selectable fields include the various timeout intervals as well as synchronization modes.
- Row Cache Dashboard** The “Row Cache Dashboard” screen contains the global database row cache information that can be dynamically modified. Selectable fields include the various row cache thresholds and working set information.
- RUJ Dashboard** The “RUJ Dashboard” screen contains global database recovery-unit journal information that can be dynamically modified. Selectable fields include the journal allocation and extension sizes.
- Monitor Dashboard** The “Monitor Dashboard” screen contains the global database I/O information that can be dynamically modified. Selectable fields include the maximum number of Database Recovery (DBR) processes that will be invoked following node failure, the default priority that the detached ABS server process will be invoked with, the default priority that the detached ALS server process will be invoked with, DBR server process will be invoked with, the detached LCS server process will be invoked with, the detached LRS server process will be invoked with and the detached RCS server process will be invoked with
- Obviously, node failure in this screen refers to another node failing.
- In addition, once a server has been invoked, you can only change the servers' priority using the Tools menu.
- ABS Dashboard** The “ABS Dashboard” screen contains global database AIJ Backup Server (ABS) process information that can be dynamically modified. Selectable fields include several boolean attributes such as quiet-point backup status.
- ALS Dashboard** The “ALS Dashboard” screen contains global database AIJ Log Server (ALS) process information that can be dynamically modified. Selectable fields include several group-commit thresholds.
- DBR Dashboard** The “DBR Dashboard” screen contains global Database Recovery (DBR) process information that can be dynamically modified. Selectable fields include buffer-related attributes.

RCS Dashboard The “RCS Dashboard” screen contains global Row Cache Server (RCS) process information that can be dynamically modified. Selectable fields include checkpoint buffer size information as well as sweep thresholds.

Per-Process Dashboard The “Per-Process Dashboard” screen contains the information that can be dynamically modified for a specific process database attach. When selecting this screen, you will be presented with a menu of the active processes on this node, or you can manually enter a specific process identifier.

RUN - TIME
 DATABASE
 DASHBOARD
 MODIFICATION

Use the “Update” on-screen menu option to modify a particular field. You will be presented with a menu to identify the field you wish to update. You will then be asked to enter the new value for this particular field. You may use “Ctrl/Z” at any time to cancel input. It's that easy.

The updates made to any fields are not stored in the database. The purpose of updating these fields is to test, monitor and analyze the effects of these changes on the database, so that you can later make persistent changes to appropriate database attributes using SQL.

The “Config” on-screen menu contains options that allow you to select active or passive notification. Active notification means that all processes attached to the database on that node will be immediately notified that a database attribute setting has been modified and that they should reset their individual parameter values. Passive notification means that the user processes will be notified passively at intervals determined by the database software. Passive notification is the default, as it is non-intrusive to system performance.


The advantage of passive notification is that there is no impact on the system. The disadvantage is that you have no control over when the individual processes will notice that a particular database parameter setting has changed.


The advantage of active notification is that each database user on the node is immediately notified of the change and that the new parameter settings are immediately used. The disadvantage is that system performance may be temporarily impacted while the processes respond to the notification.


When using passive notification, a broadcast notification can be performed using the “Notify users of previous changes” option of the configuration submenu.


RULES AND
 RESTRICTIONS

The following rules and restrictions apply to the “Database Dashboard” facility:


 On OpenVMS, you must have WORLD and BYPASS privilege to be able to dynamically update any dashboard screen. On Digital UNIX, no updating is currently permitted.


 When migrating amongst the “Database Dashboard” screens using the right-arrow (\Rightarrow), next screen and left-arrow (\Leftarrow) previous screen keys, you cannot visit the “Per-Process Dashboard” screen.


 When visiting the “Per-Process Dashboard” screen, if the process being examined detaches from the database, you will be automatically positioned on the “IO Dashboard” screen.

 Changes made by other RMU Show Statistic users dynamically modifying the database are reflected in the “Current Value” column only. The other five columns reflect changes made by the current user only.

 The “Database Dashboard” is available in online mode only; it is not available during binary file replay.

 All database attributes are updated only on the current node. Other nodes are unaffected and unaware of any dynamic changes made on other nodes.

 The “Database Dashboard” screens *are* written to the STATISTICS.RPT file created by the “Options” on-screen menu option.

 Updated database parameter settings are not updated in the database rootfile (.RDB). Once the database is closed, the current settings are lost forever.

WARNINGS

It should be noted that there is no error checking or validation of the “updated” field values with respect to their interactions on the proper operation of the database. The modified values are checked for proper range criteria.

This means that the DBA is free to cause all kinds of havoc if an improper value is entered in an inappropriate field. The dashboard is a powerful tool but, like any powerful tool, it can be misused.

Online Analysis

Chapter 10 - Online Analysis

INTRODUCTION

There are three distinct and iterative phases to performance tuning of a database: monitoring, analysis, and correction. These phases are iterative and on-going. Monitoring of database performance leads to analysis of detected problems. Analysis leads to corrective action that you will need to monitor closely, thereby restarting the database tuning process.

Most database products provide tools for the purpose of collecting database statistics and monitoring the performance characteristics of the application. However, these tools seldom provide integrated realtime analytical tools that enable the DBA to identify potential performance problems.

The role of the database administrator has evolved from just being the database maintainer into being the performance analyst.

The RMU Show Statistic utility provides an “Online Analysis” facility. The facility identifies possible performance problems that have the potential to be of interest to a DBA. This “Online Analysis” facility uses the runtime statistic information being collected to provide the DBA with areas in which additional monitoring may be necessary.

DISCLAIMER

The “Online Analysis” facility information is not a substitute for the vital services performed by professional database administrators and consultants, but serves only as a source of additional information. The observations and recommendations contained in this document use generally accepted database performance tuning guidelines. The recommendations may not always apply to your specific circumstances. Only a trained consultant or database administrator can accurately diagnose and make database parameter recommendations for your specific situation.

ONLINE ANALYSIS
FACILITY
OVERVIEW

You access the facility by selecting the “Online Analysis & Info.” option from the main menu of the RMU Show Statistic utility.



The items identified in the Online Analysis screens are not necessarily indications of performance problems. The items may be an indication of a performance problem, but in most cases, further investigation and research is necessary.

FACILITY
AVAILABILITY

The “Online Analysis” facility screen information is not available in the binary output file produced using the OUTPUT qualifier. Consequently, the Online Analysis facility is not available when you replay a binary file using the INPUT qualifier.

The reason for this restriction is that the Online Analysis facility reviews certain database parameters and attributes as part of the analysis phase, and access to this information is not available during binary file replay.

HOW ONLINE
ANALYSIS WORKS

Each screen in the “Online Analysis” facility performs a series of calculations and algorithms that relate to a common database component, such as buffers, indexes or after-image journals. Most calculations or algorithms have a corresponding threshold, or expected upper or lower bounds of the calculation. You can change these thresholds using either logical names, configuration variables, or at runtime. If a particular result from a calculation exceeds the established threshold, the facility will display an informational message on the screen.

Under ideal circumstances, the “Online Analysis” facility will not detect any calculations or algorithms exceeding the corresponding thresholds for the particular component. When this happens, the facility will display the message “Nothing to report with current configuration” to notify you of this condition.

The “Online Analysis” facility uses the actual database parameters and attributes specified using interactive SQL. The “Online Analysis” facility does not use the “Database Dashboard” facility because the dashboard attributes are temporary settings. Therefore, you can make online changes using the “Database Dashboard” facility, but these changes do not impact the analysis output.



However, Oracle Corporation recommends that you use the “Database Dashboard” facility for testing potential database attribute settings as this is the easiest method available for testing your changes. You must use interactive SQL in order to make the attribute settings persistent.

The “Online Analysis” facility operates at the designated screen refresh interval. The “Online Analysis” facility is fairly CPU-intensive, and the analysis results seldom vary greatly over a minuscule period of time. Therefore, Oracle Corporation recommends

that the screen refresh interval normally be set to 3 seconds or more. Using a screen refresh rate of 1 second or less may degrade application throughput.

ONLINE ANALYSIS
ORIENTATION



The “Online Analysis” facility algorithms tend to bias themselves towards high-throughput applications. These types of applications are typically the least understood in terms of transaction flow and impact on system resources.

Note that the “Online Analysis” facility does not attempt to perform any kind of operating system specific analysis; the operating system typically provides tools much better suited to this activity.

ONLINE ANALYSIS
THRESHOLDS

Most of the “Online Analysis” facility calculations have corresponding values, or thresholds, which determine if an alert message should be displayed. You can change these thresholds using either logical names, configuration variables, or at runtime.

You can specify a threshold before starting the RMU Show Statistic utility using logical names. All online analysis logical names have the logical name format `RDM$BIND_STATS_XXX`, where the `XXX` identifies the specific threshold being defined.

You can also specify thresholds before starting the RMU Show Statistic utility using configuration variables, which reside in the RMU Show Statistic utility configuration file for the database. All online analysis variables have the same name `XXX` that corresponds to the logical name described above.

You may define thresholds at runtime using the options available in the “Config” on-screen submenu located on most Online Analysis facility screens.

You can save thresholds changed at runtime by exporting a new configuration file, using the Tools menu. You obtain the Tools menu by typing the exclamation point (!).

CLUSTER
STATISTIC
COLLECTION

The Online Analysis facility is an integral component of the Cluster Statistic Collection and Presentation facility. This means that, when cluster statistic collection is active, the online analysis facility is able to analyze the complete database performance on all nodes of the cluster, in realtime. This is an extremely valuable benefit to the DBA; no other database product provides this type of capability.

LOGICAL AREA
ANALYSIS

The “Online Analysis” facility does not currently perform analysis at the logical area level. A logical area is a database object such as a table, a b-tree index, a hash index or a blob. All analysis uses the summary statistic information.

Support for logical area analysis may be available in a future release.

ANALYSIS
MESSAGE TYPES

The “Online Analysis” facility contains two basic types of messages: attribute anomalies and threshold anomalies.

The facility displays attribute anomaly messages when it detects enabled database attributes that may cause performance or integrity problems. These messages are also displayed when disabled database attributes should be enabled to improve database performance.

The facility displays threshold anomaly messages when a specified threshold is exceeded by some algorithmic analysis.

ONLINE ANALYSIS
MESSAGES

The following sections describe the various messages displayed by the “Online Analysis” facility, and provides detailed explanations about the causes and actions required to address the messages.

BUFFER ANALYSIS
SCREENS

The “Buffer Analysis” screen provides information about local and global buffer performance. You can use the information in this screen to examine the effectiveness of local and global buffers.

Asynchronous pre-fetch is disabled

This message indicates that the Asynchronous Pre-Fetch (APF) feature is not available for your database. The APF feature reduces the amount of time that a process waits for page retrievals from the storage area by anticipating which pages will need to be available next. These anticipated pages can then be retrieved before the application actually requests the pages.

When the APF feature is available, Oracle Rdb examines each process’ operations and attempts to predict the process’ future access patterns; the APF feature identifies sequential scans. When Oracle Rdb detects a sequential scan being performed, it fetches the pages that it anticipates the application will subsequently request. This pre-fetching of pages is asynchronous.

The APF feature is available by default. You can also use the `RDM$BIND_APF_ENABLED` logical name to enable or disable the APF feature. The value “0” will disable the feature and the value “1” will enable the feature.



Oracle Corporation recommends that you use SQL syntax to enable or disable the APF feature.

ected asynchronous pre-fetch is disabled

This message indicates that the Detected Asynchronous Pre-Fetch (DAPF) feature is not available for your database. The DAPF feature reduces the amount of time that a process waits for page retrievals from the storage area by anticipating which pages will need to be available next. These anticipated pages are then subsequently retrieved before the application actually requests the pages.

When the DAPF feature is available, Oracle Rdb examines the operations of each process and attempts to predict the future access patterns of the application. The DAPF feature is able to detect more than just the identification of sequential scans. It applies artificial intelligence heuristics to predict access patterns, even within line-item-type clusters. This pre-fetching of pages is asynchronous.

The DAPF feature is available by default. You can also use the `RDM$BIND_DAPF_ENABLED` logical name to enable or disable the DAPF feature. The value “0” will disable the feature and the value “1” will enable the feature.



Oracle Corporation recommends that you use SQL syntax to enable or disable the DAPF feature.

nchronous batch-write is disabled

This message indicates that the Asynchronous Batch-Write (ABW) feature is not available for your database. The ABW feature reduces the number of stalls experienced by database processes while waiting for write I/O operations to the storage areas to complete.

Earlier versions of Oracle Rdb performed page-write operations in batches. Whenever a data page needed to be written back to the storage area, a set of other modified data pages would also be gathered together and written to disk in parallel. However, the process still had to wait until all of the disk writes completed. This is essentially a synchronous write I/O operation, with the result that the process could not perform additional application work until all write operations completed. Also, the impact of these write I/O operations on the disk device and disk controller is quite often intrusive, requiring additional resources to manage the sudden increase in I/O activity.

The goal of the ABW feature is to increase database performance by making it possible for a certain number of buffers in the allocate set of each process to have disk write I/O operations in progress at any time without requiring the process to wait for

any of those write I/O operations to complete. For each individual process, performance is optimal when a constant and sustainable series of disk write I/O operations are being performed. This allows the disk controller to manage I/O queues and disk-head movement in a much more controlled manner, thereby improving overall system performance for all database users.

The ABW feature is very important when using the Fast Commit feature, because checkpoints cause the entire set or portion of modified pages for a process to be written asynchronously back to the database storage areas.

The Fast Commit feature improves database throughput by delaying buffer write I/O operations as long as possible. This improves the re-use of hot pages and eliminates unnecessary write I/O operations. However, one of the major performance problems of the Fast Commit feature is being blasted for one of these modified pages by another user, and especially being blasted by the Database Recovery (DBR) process, DBR. Your process essentially stops everything, synchronously writes the record modifications to the RUJ file, sometimes synchronously writes the SPAM page and finally writes the affected data page to disk and releases the page lock. You can avoid this by pre-writing the pages asynchronously, including the RUJ file, using the ABW feature. However, proper settings are necessary.



Oracle Corporation recommends that you review the *Guide to Database Performance and Tuning* for additional information.



Note that it is difficult to recommend an ideal setting because each application is different. In some cases, it is sufficient to maintain your buffer pool 50% at empty, but sometimes 75% may be a better value; you need to decide. A further complication is that the DBR process prevents your process from doing application work due to the database freeze. Therefore, DBR requesting pages sometimes results in your process receiving “deadlock on freeze” on pages being requested by the DBR process. Even though these deadlocks are internal they still wait for the interval specified by the OpenVMS SYSGEN DEADLOCK_WAIT parameter. The DEADLOCK_WAIT SYSGEN parameter defaults to 10 seconds; Oracle highly recommends that you change this value to one or two seconds at most.

The ABW feature is available by default. You can also use the RDM\$BIND_ABW_ENABLED logical name to enable or disable the ABW feature. The value “0” will disable the feature and the value “1” will enable the feature.



Oracle Corporation recommends that you use SQL syntax to enable or disable ABW feature.

'AS <n.m>% page hit below <x.y>% threshold

This message indicates that the allocate set effectiveness is less than the corresponding threshold. This message applies to both local and global buffers; when using local buffers, the buffer pool is the allocate set.

The percentage is computed as the total number of pages found in the local buffer or global buffer allocate set, divided by the total number of page requests. In general, a higher percentage implies better buffering and, therefore, fewer stalls and better throughput. If every page request found the page in the buffer pool, the buffer pool effectiveness would be 100%.

However, an allocate set effectiveness of 100% is clearly impossible, since your process attaches to the database with an empty buffer pool. Therefore, at least one page request will result in the page not being found in the buffer pool.

Also, note that when using global buffers, this analysis is measuring the allocate set size instead of the overall global buffer pool effectiveness. It is quite possible that the global buffer pool effectiveness is adequate while the allocate set effectiveness is not. Typically, reducing the number of attached users or increasing the number of buffers improves the allocate set effectiveness.

The default allocate set effectiveness threshold is 75%. You can override the default value with the `RDMSBIND_STATS_LB_PAGE_HIT_RATIO` logical name, the `LB_PAGE_HIT_RATIO` configuration variable or by selecting the “LB/AS page hit threshold” option from the “Config” on-screen menu.

<n.m>% pool hit below <n.m>% threshold

This message indicates that your global buffer pool effectiveness is less than the corresponding threshold.

The percentage is computed as the total number of pages found in the global buffer pool, divided by the total number of page requests. In general, a higher percentage implies better buffering and, therefore, fewer stalls and better throughput. If every page request found the page in the buffer pool, the buffer pool effectiveness would be 100%.

However, a global buffer pool effectiveness of 100% is clearly impossible, since the monitor opens the database with an empty global buffer. Therefore, at least one page request will result in the page not being found in the buffer pool.

The default local buffer pool effectiveness threshold is 85%. You can override the default value with the `RDMSBIND_STATS_GB_POOL_HIT_RATIO` logical name, the `GB_POOL_HIT_RATIO` configuration variable or by selecting the “GB pool hit threshold” option from the “Config” on-screen menu.

<n.m>% IO-saved below <n.m>% threshold

This message indicates that the percentage of page requests that saved a disk I/O is less than the corresponding threshold. This message is very useful in determining whether local buffers are more efficient than global buffers.

The saved I/O percentage is computed as the total number of pages requested divided by the total number of pages found. In general, a higher percentage implies a better benefit which you can obtain from using global buffers. As the percentage approaches 0%, you may want to consider using local buffers. Conversely, as the percentage approaches 100%, you may want to continue using global buffers.

Adjusting the size of the global buffer affects this value. The larger the global buffer, the higher likelihood of being able to save read I/O operations.

The default local buffer pool effectiveness threshold is 85%. You can override the default value with the `RDMSBIND_STATS_GB_IO_SAVED_RATIO` logical name, the `GB_IO_SAVED_RATIO` configuration variable or by selecting the “GB IO-saved threshold” option from the configuration submenu.

TRANSACTION
 ANALYSIS SCREEN

The “Transaction Analysis” screen provides information about transaction performance. Primarily, this screen examines average transaction duration.

m>% verb rollback above <n.m>% threshold

This message indicates that the SQL verb rollback percentage has exceeded the corresponding threshold.

The percentage is computed by dividing the number of verb failures by the total number of verbs.

A SQL verb is an atomic statement or action. For example, a record insert is a verb, as is a record deletion.

Excessive verb failures are usually an indication of a failed constraint, such as uniqueness criteria, or an invalid DDL statement. Note that in the case of cursors and scans, reaching the end-of-stream always results in a verb failure.

Note that SQL performs its own internal queries to identify metadata, such as table or index names.

Also, within a compound statement, each individual statement is atomic and Oracle Rdb performs a verb-success operation after processing each one. To avoid this overhead, you can use the SQL BEGIN ATOMIC statement to treat the entire block as a single verb.

Oracle Rdb rarely issues a verb failure unless there is an exception of some kind, such as a constraint failure.

The default verb rollback threshold is 25 percent. You can override the default value with the RDM\$BIND_STATS_VERB_SUCCESS_RATIO logical name, or by defining the VERB_SUCCESS_RATIO configuration variable, or by selecting the “Verb success threshold” option from the configuration submenu.

>th %ile transaction duration: <n.m> seconds

This message indicates that the nth percentile aggregate transaction duration has exceeded the corresponding threshold. The indicated number of seconds represents the nth percentile time, not the specified threshold. Note that this message is for both read/write and read-only transactions combined; there are separate messages for just read/write and read-only transaction durations.

The duration of each transaction measures the first SQL SET TRANSACTION statement through the completion of the transaction with a COMMIT or ROLLBACK statement. As each transaction completes, its duration increases the cumulative value maintained by the database.

This message is very useful for eliminating the infrequent long-duration transaction from the determination of the typical transaction duration.

The default transaction duration percentile is 95%. You can override the default value with the RDMSBIND_STATS_MAX_TX_PERCENTILE logical name, or by defining the MAX_TX_PERCENTILE configuration variable, or by selecting the “Transaction duration percentile” option from the configuration submenu.

The default transaction duration threshold is 15 seconds. You can override the default value with the RDMSBIND_STATS_MAX_TX_DURATION logical name, or by defining the MAX_TX_DURATION configuration variable, or by selecting the “Transaction duration threshold” option from the configuration submenu.

>th %ile read/write transaction duration: <n.m> seconds²²

This message indicates that the nth percentile read/write transaction duration has exceeded the corresponding threshold. The indicated number of seconds represents the nth percentile time, not the specified threshold. Note that this message is for read/write transactions only; there are separate messages for aggregate and read-only transaction durations.

The duration of each transaction measures the first SQL SET TRANSACTION statement through the completion of the transaction with a COMMIT or ROLLBACK statement. As each transaction completes, its duration increases the cumulative value maintained by the database.

This message is very useful for eliminating the infrequent long-duration transaction from the determination of the typical transaction duration.

The default transaction duration percentile is 95%. You can override the default value with the RDMSBIND_STATS_MAX_RW_TX_PERCENTILE logical name, or by defining the MAX_RW_TX_PERCENTILE configuration variable, or by selecting the “Read/Write Transaction duration percentile” option from the configuration submenu.

The default transaction duration threshold is 15 seconds. You can override the default value with the RDMSBIND_STATS_MAX_RW_TX_DURATION logical name, or by defining the MAX_RW_TX_DURATION configuration variable, or by selecting the

²² Oracle Rdb 7.0.2.0

“Read/Write Transaction duration threshold“ option from the configuration sub-menu.

>th %ile read-only transaction duration: <n.m> seconds²³

This message indicates that the nth percentile read-only transaction duration has exceeded the corresponding threshold. The indicated number of seconds represents the nth percentile time, not the specified threshold. Note that this message is for read-only transactions only; there are separate messages for aggregate and read/write transaction durations.

The duration of each transaction measures the first SQL SET TRANSACTION statement through the completion of the transaction with a COMMIT or ROLLBACK statement. As each transaction completes, its duration increases the cumulative value maintained by the database.

This message is very useful for eliminating the infrequent long-duration transaction from the determination of the typical transaction duration.

The default transaction duration percentile is 95%. You can override the default value with the RDMSBIND_STATS_MAX_RO_TX_PERCENTILE logical name, or by defining the MAX_RO_TX_PERCENTILE configuration variable, or by selecting the “Read-Only Transaction duration percentile“ option from the configuration submenu.

The default transaction duration threshold is 15 seconds. You can override the default value with the RDMSBIND_STATS_MAX_RO_TX_DURATION logical name, or by defining the MAX_RO_TX_DURATION configuration variable, or by selecting the “Read-Only Transaction duration threshold“ option from the configuration submenu.

²³ Oracle Rdb 7.0.2.0

AIJ ANALYSIS
 SCREEN

The “AIJ Analysis” screen provides information about after-image journal performance. You can use the information in this screen to ensure that after-image journals are on their own device. You can also examine the AIJ request block to I/O ratio, blocks per I/O ratio, and the formatting of the AIJ request block.

After-image journaling is disabled

This message indicates that the after-image journaling feature is not available. By default, a newly created database or a newly restored database does not use the after-image journaling facility. When after-image journaling is not available, the database is not recoverable in the event of media failure, such as a disk crash.

Using the after-image journaling facility without the Fast Commit feature usually incurs about a 10% reduction in performance. However, when combined with the Fast Commit feature, a performance improvement of up to 50% is often possible. The Commit-to-Journal feature is able to improve performance even more. These types of features rely on the after-image journal for recovery in the event of process failure.


Database should be configured for at least <n> journals

This message indicates that the database uses either a single extensible after-image journal, or two fixed-size after-image journals. Both of these combinations provide less than optimal performance characteristics.

Extensible after-image journals are excellent when you cannot predict transaction workloads, or you are short on disk space and cannot afford to pre-allocate multiple fixed-size after-image journals. Extensible after-image journals will extend automatically as they become full. However, all transaction activity suspends during the AIJ extension operation, on all database nodes.

The AIJ extension operation performs at a maximum rate of 2,000 blocks per second. For example, a 10,000 block extension takes a minimum of five seconds to complete, even under optimal circumstances. This pause can be quite noticeable on OLTP-type applications where the suspension of transactional activity directly impacts throughput.

Fixed-size after-image journals eliminate the extension problem by providing multiple ready-to-use after-image journals. When one after-image journal fills up, you simply switch to another available after-image journal. However, fixed-size journals must be available in order to be accessible. This means that you must back up the old journals faster than the application can fill them. This is not often possible with just two fixed-size journals.


 Therefore, Oracle Corporation recommends that you always have a minimum of three fixed-size journals and, preferably, at least five. Oracle also recommends that you always use the AIJ backup server to automate the AIJ backup operation.

The value <n> is based on the number of reserved after-image journal slots.


at least <n> journals should be reserved

This message indicates that the number of available after-image journal slots is less than two. AIJ slots are pre-allocated by the DBA, and determine the maximum number of after-image journals that can be created for a database.

With zero or one AIJ slots available, your database is at risk of suspending during an AIJ switch-over operation.

 Oracle Corporation recommends that your database always have at least two available after-image journal slots and, preferably, at least five.


Log server is manual This message indicates that the database monitor does not start the AIJ Log Server (ALS) unless specifically directed by the DBA. There is no threshold associated with this message.

 Oracle Corporation recommends that the ALS always be invoked automatically.

When the database monitor does not invoke the ALS process automatically, application processes need to acquire a local lock for their own updated buffers, as well as the updated buffers of other processes, before doing a group-commit of the updated buffers to disk. When high update activity is occurring on the database, this lock contention can cause a decrease in performance and system throughput.

Also, when using circular after-image journals, the ALS process is able to automatically create emergency after-image journals in the event of an AIJ switch-over suspension.

Log server is automatic This message indicates that the database monitor starts the AIJ Log Server (ALS) automatically at database open time. There is no threshold associated with this message.

 Oracle Corporation recommends that the ALS process always be invoked automatically.

Journal overwrite is enabled This message indicates that the after-image journals can be re-used without having first been backed up. This attribute is only useful for those databases that can be re-created from scratch, as they are not recoverable.



Oracle Corporation recommends that the journal overwrite feature always be disabled. Use of the emergency after-image journal facility is a preferable method of ensuring continued application processing in the event of suspended AIJ switch-over operations.

throughput <n.m> excessive for allocation <n> blocks (minimum <m>)

This message indicates that the AIJ throughput will cause the current after-image journal to extend or switch-over in a time period that is less than the corresponding threshold.

AIJ Request Blocks (ARB) are pieces of information to be written to the after-image journal. Each ARB is approximately 2k in size but often contains less information than that. In addition, ARBs contain different types of data; for instance, the most common ARB type is “data” but ARBs can also contain “commit” information, “rollback” and “checkpoint” information, and a host of other interesting types.

Each process initializes an ARB and submits it to be formatted by the group commit process. If you are not using the AIJ Log Server (ALS), one of the application processes becomes the group commit process. If you are using the ALS, then the ALS is always the group commit process. Using the ALS is desirable for several reasons. The first reason is that formatting the ARBs is CPU-intensive and allowing application processes to do this means they are NOT doing application work. The second reason is that the ALS formats ARBs and nothing else; it uses a highly optimized algorithm, using a double-buffering scheme to perform “recoverable asynchronous I/O” to the after-image journal.

Note: Oracle Rdb is the only production database that allows asynchronous I/O to the after-image journal.

The ALS algorithm always attempts to format more ARBs while the previously formatted ARBs are being written to disk. A foreground format occurs when ARBs are formatted while no after-image journal write I/O operations are pending. A background format occurs when ARBs are formatted while after-image journal write I/O operations are pending. Ideally, it is desirable to format all ARBs in background since this results in the best utilization of the ALS process. Remember that the after-image journal is a sequential file. It is extremely desirable to keep the data streaming to the disk so that disk-head movement is optimal. Requiring the ALS to stop and format ARBs then issue an I/O means that the disk head is not properly positioned for optimum efficiency, with the result of increased latency.

Suppose 8.9% of your ARBs are formatted in the background. This means that 91.1% of the time, you are formatting ARBs while no after-image journal disk I/O is pending. This results in longer after-image journal I/O duration, which typically impacts transaction commit operation durations.

What are the causes of this? Typically, this means that your transactions commit intermittently (most frequent reason), or they update relatively little information. Using the Dashboard facility of the RMU Show Statistic utility is the easiest method available for experimenting with various database attribute adjustments. Try adjusting the “Max IO Blocks” dashboard entry lower. Possibly using the Fast Commit feature will help improve this event. The interesting use of this message is in determining whether the expense of using the ALS process is worth the benefit. If you have almost no background ARB formats, you might be better off not using the ALS process.

The default AIJ seconds to extend threshold is 60 seconds. You can override the default value with the `RDMSBIND_STATS_AIJ_SEC_TO_EXTEND` logical name, or by defining the `AIJ_SEC_TO_EXTEND` configuration variable, or by selecting the “Seconds to AIJ extend/switch” option from the configuration submenu.

AIJ backup server is manual This message indicates that the database monitor will not automatically invoke the AIJ Backup Server (ABS) on an after-image journal switch-over operation.

Failure to perform a timely backup of the after-image journals may cause the after-image journal switch-over operation to suspend. During the suspension, all database transactional activity suspends until an available after-image journal can be found.



Oracle Corporation recommends the automatic invocation of the ABS process when using circular after-image journals.

; cannot be invoked because only 1 journal is active

This message indicates that the database monitor will invoke the AIJ Backup Server (ABS) automatically. However, the database monitor is unable to invoke the ABS process because extensible after-image journals are being used.

The database monitor only invokes the ABS process following an after-image journal switch-over operation. Since extensible after-image journals are unable to switch over, the database monitor will not invoke the ABS.



Oracle Corporation recommends that you use fixed-size (circular) after-image journals.

↳ cannot be invoked because a journal has been overwritten

This message indicates that the database monitor will invoke the AIJ Backup Server (ABS) automatically. However, the database monitor is unable to invoke the ABS process because one or more of the fixed-size after-image journals have been overwritten. Over-writing the previous contents of the after-image journal is destructive, and the recovery information cannot be re-obtained.



Oracle Corporation recommends that you immediately back up your database as it cannot be recovered in the event of media failure.

↳ cannot be invoked because hard data loss has occurred

This message indicates that the database monitor invokes the AIJ Backup Server (ABS) automatically. However, the database monitor is unable to invoke the ABS process because one or more of the fixed-size after-image journals have experienced hard data loss.

Hard data loss occurs when a process is unable to write some AIJ modifications to any after-image journal and the information must be discarded. This typically occurs when the disk device on which the after-image journals reside is not accessible in some manner. The database is always shut down following hard data loss to preserve integrity. However, the database can be restarted once new after-image journals have been made available to the system.



Oracle Corporation recommends that you immediately back up your database as it cannot be recovered in the event of media failure.

↳ cannot be invoked because backup file name has not been specified

This message indicates that the database monitor invokes the AIJ Backup Server (ABS) automatically. However, the database monitor is unable to invoke the ABS process because none of the after-image journals have a valid backup file name specification.

You can define the after-image journal backup file name specifications either globally for all after-image journals, or on a per-journal basis.

<AIJ_name> device <device_name> same as storage area

This message indicates that the named after-image journal resides on a disk device that one or more storage areas share.

Oracle highly recommends that after-image journals reside on their own disk device. This eliminates a double point-of-failure. For example, if the after-image journal were to reside on the same device as a storage area, then losing that disk device would jeopardize both the after-image journal and the storage area; you would be unable to recover the storage area because you also lost the source of the recovery information.

ARB pool exhausted <n> times This message indicates that the AIJ Request Block (ARB) pool has been fully consumed the indicated number of times.

By default, there are a maximum of 300 ARBs per database per node. All application users on the particular node share these ARBs. When the application processes consume all of the ARBs, all transaction processing stops until the group commit process formats some ARBs and makes them available to the application processes.

Running out of available ARBs is an extremely serious situation; it should always be avoided, if possible. The most frequent cause of this event is that the number of application processes greatly exceed the number of available ARBs. Another cause is transactions that update a large amount of data at once, such as bulk loads or mass updates.

In Oracle Rdb 7.0.1.0, the RMU Show Statistic utility displays the number of available ARBs on the “AIJ Journal Information” screen. Oracle Rdb allows you to increase the maximum ARB value up to 8,192 using the “ARB Count” dashboard entry, located in the “AIJ Dashboard.” Oracle Rdb also allows you to increase the maximum ARB value up to 8,192 using the RDM\$BIND_AIJ_ARB_COUNT logical name. There is no configuration variable to control the number of ARBs.

m>% ARBs per I/O below <n.m>% threshold

This message indicates that the average number of AIJ Request Blocks (ARB) formatted for each after-image journal I/O operation is less than the corresponding threshold.

This message is computed by dividing the total number of ARBs submitted by the total number of after-image journal write I/O operations.

As a process submits each ARB, the group commit process formats them into a single I/O buffer and ultimately writes the completed buffer to the after-image journal. Ideally, there should be many ARBs per after-image journal I/O, since this allows more transactional modifications to be written more efficiently.

Conversely, an extremely large number of ARBs per after-image journal I/O may be an indication of small transactional modifications performed by many processes simultaneously. This usually results in sub-optimal ARB utilization, typically resulting in the ARB pool being frequently exhausted.

An ARB can contain approximately 2k in transaction modification information. The after-image journal buffer size contains a maximum of 63k in transaction modification information. Therefore, an optimal number of ARBs per after-image journal I/O operation is in the 30 to 32 range. However, it is extremely difficult to achieve this goal in real life. Therefore, a more reasonable number of ARBs is in the 20-25 range.

The default ARBs per I/O threshold is 2 ARBs. You can override the default value with the `RDMSBIND_STATS_AIJ_ARBS_PER_IO` logical name, or by defining the `AIJ_ARBS_PER_IO` configuration variable, or by selecting the “ARBs per AIJ I/O” option from the configuration submenu.

m>% blocks written per I/O below <n.m>% threshold

This message indicates that the average number of blocks written per after-image journal write I/O operation is less than the corresponding threshold.

This message is computed by dividing the total number of after-image journal blocks written by the total number of after-image journal write I/O operations.

There is a maximum number of 127 blocks written to the after-image journal per write I/O operation. Ideally, you want the average number of blocks to be as close to that value as possible.

The AIJ Log Server (ALS) is extremely efficient at issuing as many write I/O operations to the after-image journal as possible. By doing this, the ALS process is able to keep the data streaming to the AIJ disk device. However, this sometimes sacrifices performance because more I/O is being written than is really necessary.

If the averaging number of blocks per after-image journal write I/O operation is small, you can increase the minimum group commit buffer size using the `RDMSBIND_AIJ_IO_MIN` logical name. There is no configuration variable for this value.

The default blocks per I/O threshold is 2 blocks. You can override the default value with the `RDMSBIND_STATS_AIJ_BLK_PER_IO` logical name, or by defining the `AIJ_BLK_PER_IO` configuration variable, or by selecting the “Blocks per AIJ I/O” option from the configuration submenu.

m>% background-formatted ARBs below <n.m>% threshold

This message indicates that the average number of AIJ Request Blocks (ARB) formatted in background by the AIJ Log Server (ALS) is less than the corresponding threshold.

The percentage is computed by dividing the total number of ARBs formatted in background by the total number of ARBs formatted (foreground and background).

Using the ALS is desirable for several reasons. The primary reason is that formatting the ARBs is CPU-intensive and allowing application processes to do this means they are not doing application work. The secondary reason is that the ALS formats ARBs and nothing else; it uses a highly optimized algorithm, using a double-buffering scheme to perform “recoverable asynchronous I/O” to the after-image journal.

The ALS algorithm always attempts to format more ARBs while the previously formatted ARBs are being written to disk. A foreground format results when an ARB is formatted while no disk I/O is pending. A background format results when an ARB is formatted while disk I/O is pending. Ideally, all ARBs should be formatted in background since this results in the best utilization of the ALS process. Remember that the after-image journal is a sequential file. It is extremely desirable to keep the data streaming to the disk so that disk-head movement is optimized. Having to stop and format ARBs, then issue an I/O means that the disk head is not properly positioned for optimum efficiency, with the result of increased latency.

Note that it is impossible to format 100% of the ARBs in the background, because the first after-image journal write I/O operation is always a foreground format.

The default background ARB formatting threshold is 50%. You can override the default value with the RDMSBIND_STATS_AIJ_BKGRD_ARB_RATIO logical name, or by defining the AIJ_BKGRD_ARB_RATIO configuration variable, or by selecting the “Blocks per AIJ I/O” option from the configuration submenu.

.IJ cache overflowed <n> times This message indicates that the AIJ cache has overflowed the specified number of times. There is no corresponding threshold for this message.

The AIJ cache is the buffer into which submitted ARBs are formatted. It can contain up to 63k of AIJ modification information. As ARBs are formatted into the cache, a number of events determine when a write I/O operation can be initiated to the after-image journal. In general, however, the algorithm is optimized towards keeping data streaming to the after-image journal in a steady manner. Thus, it is seldom the case that the AIJ cache becomes full before issuing a write I/O operation.

Therefore, it is a significant event when the AIJ cache overflows. This is typically indicative of excessive amounts of AIJ modifications being submitted at one time. It is possible with this type of behaviour to overwhelm the ALS into issuing a write I/O operation while an I/O is still pending. This causes the queue lengths on the disk device to become excessively large, causing a degradation of I/O throughput, which in

turn causes the AIJ cache to overflow more often. This is a catch-22 situation that should be avoided.

cache overflow above <n.m>% threshold

This message indicates that the average number of AIJ cache overflows has exceeded the corresponding threshold.

The percentage is computed by dividing the total number of AIJ cache overflows by the total number of cache format operations.

This message is useful for eliminating the occasional AIJ cache overflow occurrence from the continual cache overflow problem.

The default AIJ cache overflow threshold is 10%. You can override the default value with the `RDM$BIND_STATS_AIJ_CACHE_OVF_RATIO` logical name, or by defining the `AIJ_CACHE_OVF_RATIO` configuration variable, or by selecting the “Cache overflow threshold” option from the configuration submenu.

emergency AIJ creation not enabled

This message indicates that the “Emergency AIJ” facility is not enabled.

If the AIJ switch-over operation cannot complete because there are no available after-image journals, the database enters the “AIJ suspended” state. During this state, the DBA can add new after-image journals or perform database backups, but all other AIJ-related activities are temporarily suspended until such time as an after-image journal becomes available.

During the AIJ suspended period, any Database Recovery (DBR) process invocation causes the database to be shut down. This is required because the DBR always writes either a commit or rollback record to the after-image journal. Note that even a DBR invoked for a read-only transaction causes the database to be shut down.

The work-around to this problem is to always ensure that adequate after-image journals are available for the AIJ switch-over operation. A new logical name has been added to the Oracle Rdb: `RDM$BIND_ALS_CREATE_AIJ`. This logical name indicates whether or not the ALS server is to create an emergency after-image journal if the AIJ switch-over operation ever enters the suspended state. The default value “0” indicates that the ALS should not create an after-image journal. The value “1” indicates that the ALS should attempt to create an after-image journal. The logical name must reside in the `LNMS$SYSTEM_TABLE` logical name table.

The Database Recovery (DBR) process is also allowed to create emergency after-image journals if the logical name is enabled.

When the `RDMSBIND_ALS_CREATE_AIJ` logical name is set to the value “1”, the ALS will attempt to create an emergency after-image journal using the previous after-image journal as a template. This means that the emergency after-image journal will be created in the same directory, and with the same allocation, as the journal being switched from. If inadequate disk space, or any other error occurs, the database will simply enter the “AIJ suspended” state and the DBA must resolve the situation.

WARNING: The emergency after-image journal is not a temporary after-image journal. **DO NOT** delete it using any means other than through the standard database syntax (SQL or RMU). Manually deleting the emergency after-image journal (i.e. from DCL) will cause your database to be shut down.

The location of the emergency after-image journal (device/directory) can also be specified using the `RDMSBIND_AIJ_EMERGENCY_DIR` logical name. This logical name, if defined in the `LNMS$SYSTEM_TABLE` logical name table, should only specify the device and directory where the emergency after-image journal is to be created. Note that, if defined, the `RDMSBIND_AIJ_EMERGENCY_DIR` logical name applies to all databases on the current node. Furthermore, the logical name must not contain any non-system concealed logical name definitions.

The ALS will notify the DBA using the operator notification facility that an emergency after-image journal has been created. Furthermore, the RMU Dump Header utility will identify any after-image journal created by the ALS process.

The RMU Show Statistic utility will highlight any identified emergency after-image journal.



Note that an emergency after-image journal is a normal after-image journal in all respects. It is simply created by the ALS (or DBR) process in order to avoid the AIJ switch-over suspension state. DBR invocations due to application process failure during the after-image journal creation do not cause database shut down.

When created, the name and file specification of the emergency after-image journal is “EMERGENCY_XXX” where XXX is a series of 16 characters used to create a unique name. Note that the creation of the emergency after-image journal is not journaled. This means that the creation of an emergency after-image journal is not possible while the “Hot Standby” database replication feature is active.



Note that there is no way to remove the emergency status of an emergency after-image journal. However, an emergency after-image journal is a normal after-image journal in all other respects.

<n> journals are inaccessible This message indicates that the specified number of after-image journals are inaccessible. There is no threshold associated with this message.

An after-image journal becomes inaccessible when some event causes the after-image journal to be unreadable or unwriteable. The most frequent cause is the accidental deletion of an after-image journal.

The loss of any after-image journal is cause for concern. The database is not recoverable when it detects the loss of an after-image journal.



Oracle Corporation recommends that you immediately back up your database as it cannot be recovered in the event of media failure.

network <n.m> BPS above <n.m> BPS threshold²⁴

This message indicates that the current Hot Standby feature network transmission rate exceeds the corresponding threshold. This value is expressed in terms of bits per second.

The value is computed using the total hot standby network packet size converted into bits per second.

This message is very useful for determining when the current AIJ throughput exceeds your network capacity. Exceeding your network capacity may cause your master database throughput to degrade in order to allow the standby database sufficient time to apply the database modifications.

The default AIJ network bandwidth threshold is 60 seconds. You can override the default value with the RDM\$BIND_STATS_AIJ_NETWORK_BPS logical name, or by defining the AIJ_NETWORK_BPS configuration variable, or by selecting the “Network bandwidth threshold” option from the configuration submenu.

²⁴ Oracle Rdb 7.0.2.0

R U J A N A L Y S I S
 S C R E E N

The “RUJ Analysis” screen provides information about RUJ file performance.

m>% synchronous RUJ I/O above <n.m>% threshold

This message indicates that the average number of synchronous write I/O operations to the RUJ file exceeds the corresponding threshold.

The percentage is computed as the number of synchronous write I/O operations to the RUJ file divided by the total number of write I/O operations, both synchronous and asynchronous.

Ideally, most write I/O operations to the RUJ file should be asynchronous. Most synchronous write I/O operations to the RUJ file occur in response to a request for a particular modified page by another process. The modifications on that page must first be hardened to the RUJ file before the page can be written to the storage area and released to the requesting process. This synchronous I/O operation causes the process to stall for a relatively long time, approximately 20ms on most disk devices.

Unfortunately, there is no method available to indicate that you wish to ignore page requests from other processes.

Asynchronous write I/O operations to the RUJ file are preferable in that the process is able to continue performing application work while the I/O operations are still pending. Batch writes are the frequent cause of asynchronous write I/O operations to the RUJ file. These batch write operations typically occur at transaction commit, roll-back and checkpoint intervals, but they can also occur in response to a batch-write request.

Ideally, you want to always “pre-write” modified pages back to the database before they are requested by other users. Adjusting the asynchronous batch write facility parameters is a good method to accomplish this. In particular, you should examine the clean buffer count and batch size attributes. You can use the `RDM$BIND_BATCH_MAX` logical name, or the equivalent SQL `ABW` syntax, to specify the number of buffers a process will write asynchronously as part of the `ABW` operation. The database uses this value whenever the number of unmodified (clean) buffers for the process is less than the value specified by the `RDM$BIND_CLEAN_BUF_CNT` logical name.

The value of the `RDM$BIND_CLEAN_BUF_CNT` logical name adjusts automatically at runtime so that it does not exceed more than half the number of buffers available to the process.

It is important that the value of the `RDM$BIND_BATCH_MAX` logical name be neither too high nor too low. In general, the logical name value should probably not exceed the number of storage areas or disk devices, with the result that a sudden burst of

write I/O operation requests should not overwhelm the disk controllers. If the logical name value is too low, the process will not benefit from any disk device parallelism. For example, if you are using five disk devices but have the `RDMSBIND_BATCH_MAX` logical name set to the value “2”, then an ABW operation will write to only two disk devices in parallel.

The default RUJ file synchronous I/O threshold is 10%. You can override the default value with the `RDMSBIND_STATS_RUJ_SYNC_IO_RATIO` logical name, or by defining the `RUJ_SYNC_IO_RATIO` configuration variable, or by selecting the “Synchronous RUJ I/O threshold” option from the configuration submenu.

m>% RUJ extends above <n.m>% threshold²⁵

This message indicates that the average number of file extension operations to the RUJ file exceeded the corresponding threshold.

The percentage is computed as the number of file extension operations to the RUJ file divided by the total number of write I/O operations, both synchronous and asynchronous.

Ideally, there should be only one extend of the RUJ file per database process attach. An RUJ file is initially created with only header information; this always results in at least one file extend operation. Beyond the initial extension operation, each subsequent RUJ file extension greatly impacts the transaction throughput.



Oracle Corporation recommends setting the `RDMSBIND_RUJ_ALLOC_BLKCNT` and `RDMSBIND_RUJ_EXTEND_BLKCNT` logical names to a reasonable value, such that no RUJ file extends occur at runtime.

The default RUJ file extend threshold is 2%. You can override the default value with the `RDMSBIND_STATS_RUJ_EXTEND_RATIO` logical name, or by defining the `RUJ_EXTEND_RATIO` configuration variable, or by selecting the “RUJ extend threshold” option from the configuration submenu.

²⁵ Oracle Rdb 7.0.2.0

RECOVERY
 ANALYSIS SCREEN

The “Recovery Analysis” screen provides information about process failure and database recovery performance.

m>% process failure rate above <n.m>% threshold

This message indicates that the average number of Database Recovery (DBR) processes exceeds the corresponding threshold. The percentage is computed by dividing the total number of database recovery operations by the total number of database attaches.

When an application or server process terminates prematurely, the database monitor freezes all database activity and invokes a Database Recovery (DBR) process. The DBR process is responsible for ensuring that all transactional changes performed by the failed process either commit or rollback. While the DBR process is performing this recovery operation, the database monitor suspends all other transactional activity on the entire database, including on other nodes in the cluster. Note that the DBR process will recover a process using read-only transactions, even though there may not actually be anything to recover.

Ideally, there should be no application process failures, but it is often impossible to prevent external failures such as network disconnects or power outages.



In particular, Oracle Corporation recommends that you disallow the Ctrl/Y-STOP action on OpenVMS; this is a major source of process recovery operations.

The default process failure threshold is 15%. You can override the default value with the RDM\$BIND_STATS_DBR_RATIO logical name, or by defining the DBR_RATIO configuration variable, or by selecting the “DBR invocation threshold” option from the configuration submenu.

m>% process recovery duration above <n.m>% threshold²⁶

This message indicates that the average duration of the Database Recovery (DBR) processes exceeds the corresponding threshold. The percentage is computed by dividing the total durations of all database recovery operations by the total number of database recovery operations.

This message is effectively indicating the average time the database is unavailable as a result of each process failure. When an application or server process terminates pre-

²⁶ Oracle Rdb 7.0.2.0

maturely, the database monitor freezes all database activity and invokes a Database Recovery (DBR) process. The DBR process is responsible for ensuring that all transactional changes performed by the failed process either commit or rollback. While this recovery operation is being performed, the database monitor suspends all other transactional activity on the entire database, including on other nodes in the cluster. Note that the DBR process performs recovery for all failed processes, including those processes using read-only transactions, even though there may not actually be anything to recover.

Ideally, there should be no application process failures, but it is often impossible to prevent some types of external events such as network disconnects or power outages.

Database attributes such as the Fast Commit feature affect the process recovery duration. Also, features that require the Fast Commit feature often increase the process recovery duration significantly. Ideally, the process recovery duration should be as short as possible, since it affects all other processes accessing the database.

The default process failure threshold is 2 seconds. You can override the default value with the `RDMSBIND_STATS_DBR_DURATION` logical name, or by defining the `DBR_DURATION` configuration variable, or by selecting the “DBR recovery duration” option from the configuration submenu.

database backup has not been performed since <date>

This message indicates that the database has either never been backed up since it was created, or that the last database backup occurred in the past more days than the corresponding threshold.

If your database has never been backed up, Oracle highly recommends that you do so immediately. Also, never, ever use the OpenVMS backup utility to back up your database; it does not work correctly, especially when using global buffers or row caches.

Ideally, you want to back up your database every day. However, this is often impossible in today’s 7*24 environment. Therefore, Oracle Corporation recommends that you back up your database a minimum of once a week. You always need to customize a database backup strategy for your specific operating environment. The backup strategy should take into consideration the system workload, usage schedule, importance of data, and hardware environment of the database. There are some general guidelines that apply to all databases. Incorporating these guidelines into your strategy will ensure more reliable and more cost effective backups.

The following rules are essential to a proper database backup strategy:



Always use after-image journals.

- ✓ Always maintain a log of your backup schedule.
- ✓ Occasionally store backups in a separate location, preferably off-site.
- ✓ Do **not** use the OpenVMS Backup Utility on your database; Oracle even recommends that you set the do-not-backup attribute on database files.
- ✓ Spread your database across several disks. Always consider your existing hardware configuration and system workload.
- ✓ Determine the volatility and value of your data.
- ✓ Verify the integrity of your database regularly.
- ✓ Verify the integrity of your backup media after each backup.

Failure to perform these simple recommendations may result in your database being not recoverable.

The default database backup threshold is 6 days. You can override the default value with the `RDM$BIND_STATS_FULL_BACKUP_INTRVL` logical name, or by defining the `FULL_BACKUP_INTRVL` configuration variable, or by selecting the “Full database backup threshold” option from the configuration submenu.

Database is single-file²⁷ This message indicates your database is a single-file database. There are no thresholds associated with this message.

Oracle highly recommends that you convert your single-file database into a multfile database.

²⁷ Oracle Rdb 7.0.2.0

RECORD ANALYSIS
 SCREEN

The “Record Analysis” screen provides information about record storage and retrieval performance, including SPAM page retrieval performance.

m>% page discard rate above <n.m>% threshold (avg <n.m> I/Os)

This message indicates that the average number of pages discarded has exceeded the corresponding threshold, resulting in the average number of extra read I/O operations.

The percentage is computed using the total number of pages discarded divided by the total number of pages checked. The average number of read I/O operations incurred as a result of the discarded pages is computed using the total number of discarded pages that required a read I/O operation divided by the total number of pages checked.

Performance degradation most often results when storing new records in a database. Sometimes this is due to the way that Oracle Rdb manages free space on data pages. There are times when Oracle Rdb will read many pages in a storage area before it finds a page with sufficient space on it to store a record.

The most simple way to verify if any of the various pages checked scenarios described are occurring is to use the RMU Show Statistic utility. On the “Record Statistics” screen you should examine the following two statistics: “record stored” and “pages checked”. If you see that the “pages checked” number is significantly higher than the “record stored” number then it is very likely that one of the situations described in the following sections is being encountered.

Oracle Rdb uses Space Area Management, or SPAM pages to determine what database pages have space available for new records. The SPAM page is essentially a table representing a range of pages in a storage area. There is an entry in the table for each page in the range and each entry shows whether or not there is space available on the respective page. When searching for a page on which to store a record, Oracle Rdb searches for SPAM entries that show if a particular page has sufficient space available. There are conditions, however, where SPAM pages may not reflect the actual availability of space on a data page. When Oracle Rdb finds an entry that the SPAM page shows has space available on it, the page is retrieved. Oracle Rdb then examines the actual available space on the page to confirm that there is sufficient free space to store the new record. Sometimes the available space on the page may not actually be enough to store the record.

There are several reasons why there may not be adequate space on a page to store a new record. The primary reason is the setting of incorrect thresholds on a mixed format storage area or on logical area thresholds defined for a table or index. A secondary reason may be that some or all of the free space is locked for use by other database

users. Thirdly, the length used to represent the size of the data records may not represent the actual length of the data records; this is always the case for duplicate nodes on indexes that allow duplicates. Finally, the SPAM page entry may actually be incorrect.

If excessive pages checked are observed when storing records into a table that is stored in a mixed format area, or has logical area thresholds defined in its storage map, or has indexes defined that contain logical area thresholds, then the calculations used to define the thresholds should be reviewed to verify that the thresholds have been set correctly.

When a process deletes records from a database, the space allocated to the deleted records becomes reserved or locked by that process until the transaction either commits or rolls back. This ensures that the deleted records can be put back on the page if the transaction subsequently rolls back. At the time you delete the records, the SPAM page is updated to reflect the space that becomes available by deleting the records. When other database processes begin searching for available space to store records they will see in the SPAM page that there may be space available on the page where records were deleted. When that page is actually retrieved, Oracle Rdb finds that some or all of the available space on the page is locked by another process and there actually is not enough space to store another record. Therefore, you will have to select another page for storing the record, and the operation to find a page with adequate space repeats.

Typically the above scenario has little impact on database performance. However, in situations where, for example, a long-running database maintenance process that deletes records on many pages is active, there may be many database pages that have locked free space on them. This can have a significant impact on performance if there are many processes also attempting to store records in the same area at the same time the maintenance process is running.

Oracle Rdb stores the nominal length of table records in a structure called the Area Inventory Page (AIP). The length stored in the AIP that is available for use when determining how much space must be available before a page in a uniform format area is considered full. The length stored in the AIP for a record is not the actual length of the record when one or more of the following conditions exist:

- Indexes that allow duplicates.
- Tables that have been altered to add or drop columns.
- Use of segmented strings (LIST OF BYTE VARYING data type).

When an index allows duplicates, the length stored in the AIP will be 215 bytes, regardless of the actual index node size. Because an index with duplicates can have variable node sizes, the 215-byte size represents the median length of rows in the index's logical area.

When the row size in the AIP is less than the actual row length, it is highly likely that SPAM page entries will show space is available on pages when they actually have insufficient space to store another full size row. This is the most common cause of insert performance problems. For example, consider a case where an index uses a node size of 430 bytes, a common default value. The page size for the storage area where the index resides is 2 blocks. After deducting page overhead, the available space on a 2-block page is 982 bytes.

Assume that the page in this example is initially empty. A full size (430-byte) index node is stored. As 8 bytes of overhead are associated with each row stored on a page, that leaves $982 - 430 - 8 = 544$ free bytes remaining on the page. A duplicate key entry is made in that index node and thus a duplicate node is created on the same page. An initial duplicate node is 112 bytes long; duplicate nodes can have a variety of sizes depending on when they are created, but for this particular example, 112 bytes is available for use. Therefore, $544 - 112 - 8 = 424$ free bytes remain on the page.

At this point, 424 bytes remain on the page. That is greater than the 215 bytes that the AIP shows as the row length for the logical area, so the SPAM page shows that the page has space available. However, an attempt to store a full size index node on the page will fail, because the remaining free space (424 bytes) is not enough to store a 430-byte node.

In this case, you select another candidate page using the SPAM page, and the process repeats until a page that truly has sufficient free space available is found. In a logical area that contains many duplicate nodes, a significant percentage of the pages in the logical area may fit the scenario just described. When that is the case, and a new full size index node needs to be stored, many pages may need to be read and checked before one is found that can be used to store the row.



Note that the new Oracle Rdb pseudo-ranked b-tree indexes do not use duplicate nodes as described above. Oracle therefore recommends the use of pseudo-ranked b-tree indices when working on Oracle Rdb systems that are experiencing excessive page checking as a result of b-tree index duplicate nodes.

However, when using traditional b-tree indices, it is still possible to avoid the preceding scenario by using logical area thresholds. The goal is to set a threshold such that the SPAM page will show a page is full when space is insufficient to store a full size index node.

Using the previous example, here is how to properly set logical area thresholds to prevent excessive pages checked on an index with a 430-byte node size that is stored on a 2-block page. To calculate the proper threshold value to use, you must first determine how full the page can get before no more full size nodes will fit on the page. In this example, a database page can have up to $982 - 430 - 8 = 544$ bytes in use before the page is too full. Therefore, if 544 or fewer bytes are in use, then enough space remains to store another full size node. The threshold is then $544 / 982 = .553971$, or 55%.

In addition, you can determine how full a page must be before a duplicate node of 112 bytes will no longer fit. In this example, a database page can have up to $982 - 112 - 8 = 862$ bytes in use before the page is too full. Therefore, if 862 or fewer bytes are in use, then enough space remains to store another small duplicates node. The threshold is then $862 / 982 = .8778$, or 88%.



Oracle Corporation recommends that indices use an explicit STORE clause to map the index. The storage area named can be the default storage area if desired. The use of the STORE clause will ensure a separate AIP entry for each index and allow full use of the NODE SIZE and THRESHOLDS settings.

For indices allowing duplicates, logical area thresholds may be used in the index definition to prevent this scenario from occurring. Please refer to the discussion on logical area thresholds for additional information about creating and using logical area thresholds on an index.

When a table is initially created, the length of the record is stored in the AIP. If the table is later altered and additional columns are added to the table, the corresponding AIP entry is not updated with the new length of the record; the old length is still used.

The length stored in the AIP for segmented strings is always 150 bytes, regardless of the actual size of the segmented string. Thus, the same problems described above also apply to segmented strings. If a page in a uniform area has 158 (150 + 8 overhead bytes = 158) or more bytes free on it, then the SPAM entry for that page will show that there is space available. However, if an attempt is made to store a segmented string that is larger than the actual free space on that page, the page will be checked when it does not actually have sufficient free space to store the segment.

The proper way to avoid this condition is to only store segmented strings in a mixed format storage area. The storage area should have appropriate thresholds defined for the actual segment sizes being stored.

Various versions of Oracle Rdb have had problems where the SPAM page does not always accurately reflect the actual fullness of the pages in a storage area. That is, the SPAM page will indicate that there is space still available on a page when in fact there is not space available on a page, or conversely, the SPAM may show that the page is full when the page does have sufficient space remaining to store another record. The latter case will not affect insert performance, although it will cause some page space to be wasted. However, when a SPAM page shows there is space available when there really is not enough space available, it can impact insert performance. If only a few SPAM entries are incorrect, then this will not have a noticeable effect. However, if there are many SPAM page errors, there can be excessive I/O expended searching for a page with enough free space. This type of SPAM error can be detected by using the RMU Verify Area utility on the database. If there are many RMU-W-PGSPAMENT errors, then there will be pages checked that do not have enough space to store another record.

There are a few ways to correct SPAM threshold errors. Use the RMU Repair Spam utility on the database. Back up the database using the RMU Backup utility and then immediately restore the database using the RMU Restore utility. Another method is to export and import the database contents, although this is not often practical.

The default pages checked threshold is 10%. You can override the default value with the `RDMSBIND_STATS_PAGES_CHECKED_RATIO` logical name, or by defining the `PAGES_CHECKED_RATIO` configuration variable, or by selecting the “Pages checked threshold” option from the configuration submenu.

m>% SPAM page fetch rate above <n.m>% total fetched threshold²⁸

This message indicates that the average number of SPAM page fetches exceeds the corresponding threshold for the total number of SPAM page fetches.

The percentage is computed using the total number of SPAM pages fetched to store a record divided by the total number of SPAM page fetches for any reason. In most real-world applications, new records stored exceed the number of record modifications or deletions.

The default SPAM fetched threshold is 80%. You can override the default value with the `RDMSBIND_STATS_SPAM_PAG_FET_RATIO` logical name, or by defining the `SPAM_PAG_FET_RATIO` configuration variable, or by selecting the “SPAM pages fetched threshold” option from the configuration submenu.

m>% SPAM page fetch rate above <n.m>% record stored threshold²⁹

This message indicates that the average number of SPAM page fetches exceeds the corresponding threshold for stored records.

The percentage is computed using the total number of SPAM pages fetched to store a record divided by the total number of records stored.

When inserting a record into a full large uniform format area, a large number of SPAM reads is normal. Updates that cause fragmentation have the same effect when inserting the new fragment. This occurs when the logical area is scanned for free space. The SPAM pages make this much faster than reading all the data pages would be. If the logical area is full, then every SPAM page must be read to check that there is any free space. The logical area is then expanded by one clump, which contains

²⁸ Oracle Rdb 7.0.2.0

²⁹ Oracle Rdb 7.0.2.0

enough pages to fill one buffer. The next few inserts will be fast, until this new clump is full. When the current clump is exhausted, the entire area is scanned again, just in case another process may have deleted something. The average number of SPAM reads per insert should be approximately the number of SPAM pages in the area divided by the number of records that will fit in a clump.

However, the number of SPAM pages in the area will not divide evenly. All but one record insert into a clump will access just one SPAM page; the other record insert will read all SPAM pages. This fits in with only some transactions being affected by excessive SPAM page fetches.

This has been the behaviour since at least Oracle Rdb V4.0. Oracle Rdb has been enhanced to improve this behaviour in certain circumstances. As long as the SPAM fetches can be done within buffers, then the page retrieval will be fast. When the SPAM page retrievals exceed the capacity of the buffer pool, the circular nature of the read operations results in every read being done from disk. When you anticipate that records will increase in size, it is often a good idea to use a threshold for the logical area. Logical area thresholds reserve space for future expansion without subsequent record fragmentation. This is especially important when records initially contain a large number of NULL field values. Larger buffers would give larger clumps and reduce the proportion of slow inserts, but the buffer size is awkward to change. If the area used a fewer number of larger pages, then each SPAM page being bigger could cover more pages, and even fewer would be needed as the number of pages is smaller.

The default SPAM record stored threshold is 20%. You can override the default value with the `RDM$BIND_STATS_SPAM_REC_STO_RATIO` logical name, or by defining the `SPAM_REC_STO_RATIO` configuration variable, or by selecting the “SPAM records stored threshold” option from the configuration submenu.

`rea_name>` has most pages checked

This message indicates that the identified storage area contains the most pages checked. There is no threshold associated with this message; it is for informational purposes only.



Note that this message does not take into account other activity that might be occurring on the identified storage area; it simply indicates that this storage area has the most. It is quite often possible that storage areas experiencing a lot of pages checked may contain tables that should be moved to another storage area, either in their entirety or by partitioning to multiple storage areas.

`rea_name>` has `<n.m>%` synchronous pages checked

This message indicates that the identified storage area contains the largest percentage of pages checked. There is no threshold associated with this message; it is for informational purposes only.

The percentage is determined using the total number of pages checked divided by the total number of synchronous read I/O operations. The synchronous read I/O operations are used because asynchronous read I/O operations do not cause the process to stall and, therefore, are considered light-weight. However, synchronous read I/O operations cause the process to stall for approximately 20ms, during which time no other work is being performed. This makes the pages checked a disruptive operation to high throughput.



Note that this message does not take into account other activity that might be occurring on the identified storage area; it simply indicates that this storage area has the highest percentage. It is quite often possible that storage areas experiencing a lot of pages checked may contain tables that should be moved to another storage area, either in their entirety or by partitioning to multiple storage areas.

m>% fragmented records stored above <n.m>% threshold

This message indicates that the average number of fragmented records stored exceeds the corresponding threshold.

The percentage is computed using the total number of fragmented records stored divided by the total number of records stored.

Ideally, this value should be as close to 0% as possible, indicating the lack of record fragmentation.

There are several reasons why there may not be adequate space on a page to store a new record, thus causing the record to be fragmented. Thresholds can be set incorrectly on a mixed format storage area or on logical area thresholds defined for a table or index. Some or all of the free space is locked for use by other database users. The length used to represent the size of the data records does not always represent the actual length of the data records; this is always the case for duplicate nodes on indexes that allow duplicates. The SPAM page entry is actually incorrect

The default fragmented records stored threshold is 20%. You can override the default value with the RDMSBIND_STATS_RECS_STORED_RATIO logical name, or by defining the RECS_STORED_RATIO configuration variable, or by selecting the “Records stored threshold” option from the configuration submenu.

m>% fragmented records fetched above <n.m>% threshold

This message indicates that the average number of fragmented records retrieved exceeds the corresponding threshold.

The percentage is computed using the total number of fragmented records retrieved divided by the total number of records retrieved.

Ideally, this value should be as close to 0% as possible, indicating the lack of record fragmentation.

The default fragmented records fetched threshold is 20%. You can override the default value with the `RDMSBIND_STATS_RECS_FETCHED_RATIO` logical name, or by defining the `RECS_FETCHED_RATIO` configuration variable, or by selecting the “Records fetched threshold” option from the configuration submenu.

AREA ANALYSIS
 SCREEN

The “Area Analysis” screen provides information about storage area performance.

The Area Analysis screen is unique in that it does not have a configuration submenu. There are no thresholds associated with any messages in this screen.

ice <device_name> exceeds average storage area count <n>

This message indicates that the identified device contains more than the average number of storage areas. This may be an indication that the disk device contains too many storage areas and may be performing sub-optimally.

Ideally, storage areas should reside on their own disk devices. However, this may not always be possible. In those cases where spreading the storage areas over multiple disk devices is not possible, it is best to mix heavily utilized storage areas with minimal-use storage areas. Also, mixing update-intensive storage areas with read-only storage areas may be helpful.

rea_name> extended <n> times total <n> times

This message indicates that the identified storage area has extended the indicated number of times since the database monitor opened the database. The identified storage area has also extended the total number of times since the database was created.

There are several reasons why storage areas extend. The most obvious reason is that the storage area was completely full and needed to extend to allow the continued storing of new records. However, another frequent cause of storage area extension is improperly sized mixed-format storage areas used for hash indexes.

For the hash index to operate properly, sufficient space must be allocated in the storage area to ensure that all records distribute evenly and randomly across all pages of the storage area. When records begin filling pages in a non-uniform manner, performance of the hash index will degrade. The reason for this degradation is that the hashing algorithm formulates the target DBKEY from the record search key. If the target page is full, then an adjacent page will be retrieved and scanned using the SPAM thresholds you established for the storage area. Of course, using the new page will cause cascading problems when that page is subsequently selected as a target page.

Ultimately, the storage area will be extended even though the hash index may not be completely full.

rea_name> async read I/O stalls <n.m> exceeded average <n.m>

This message indicates that the average asynchronous I/O read operation duration for the identified storage area exceeded the average of all storage areas.

This message is meant to provide an indication of which storage areas reside on slow disk devices, or over-utilized disk devices. It may be necessary or prudent to move these storage areas to another disk device, if possible.

rea_name> async write I/O stalls <n.m> exceeded average <n.m>

This message indicates that the average asynchronous I/O write operation duration for the identified storage area exceeded the average of all storage areas.

This message is meant to provide an indication of which storage areas reside on slow disk devices, or over-utilized disk devices. It may be necessary or prudent to move these storage areas to another disk device, if possible.

rea_name> sync read I/O stalls <n.m> exceeded average <n.m>

This message indicates that the average synchronous I/O read operation duration for the identified storage area exceeded the average of all storage areas.

This message is meant to provide an indication of which storage areas reside on slow disk devices, or over-utilized disk devices. It may be necessary or prudent to move these storage areas to another disk device, if possible.

rea_name> sync write I/O stalls <n.m> exceeded average <n.m>

This message indicates that the average synchronous I/O write operation duration for the identified storage area exceeded the average of all storage areas.


This message is meant to provide an indication of which storage areas reside on slow disk devices, or over-utilized disk devices. It may be necessary or prudent to move these storage areas to another disk device, if possible.


LOCKING
 ANALYSIS SCREEN

The “Locking Analysis” screen provides information about locking performance. This screen is primarily concerned with examination of lock request stalls.

Process <n.m> excessive stall on lock <x> for <object>

This message indicates that the identified process has stalled on a lock for the identified object for a period of time exceeding the corresponding threshold.

 To reduce lock contention with other processes, Oracle Corporation recommends you keep your transactions as short in duration as possible, while providing as much data sharing as possible. If you need to retrieve a look-up record to display information, you should specify an appropriate sharing level that allows other transactions concurrent access to the same information.

 Oracle Corporation recommends you use the Adjustable Locking Granularity (ALG) feature. The ALG feature allows Oracle Rdb to maintain the minimum number of record locks while ensuring maximum concurrent access to those records.

Reducing the number of locks is sometimes more important than reducing the transaction duration. For example, page-level locking might be appropriate for storage areas containing clustered records. Often times, a 5:1 reduction in lock requests is possible. However, page-level locking is not always appropriate for all applications.

The default lock stall threshold is 2 seconds. You can override the default value with the `RDMSBIND_STATS_MAX_LOCK_STALL` logical name, the `MAX_LOCK_STALL` configuration variable, or the “Lock stall threshold” option from the configuration sub-menu.

Process <n.m> excessive timeouts <n> on <object>

This message indicates that the identified process has experienced excessive lock timeouts, the most recent of which was on the identified object. There is no threshold associated with this message. This message is displayed when the process’ number of lock timeouts exceeds the average of all processes’ number of lock time-outs.

Process <n.m> excessive deadlocks <n> on <object>

This message indicates that the identified process has experienced excessive lock deadlocks, the most recent of which was on the identified object. There is no thresh-

old associated with this message. This message is displayed when the process number of lock deadlocks exceeds the average of all processes number of lock deadlocks.

INDEX ANALYSIS
SCREEN

The “Index Analysis” screen provides information about B-tree and hash index performance.

m>% duplicate b-tree fetch above <n.m>% threshold

This message indicates that the average number of b-tree index duplicate node fetches exceeds the corresponding threshold.

The default b-tree index duplicate fetch threshold is 15%. You can override the default value with the `RDMSBIND_STATS_BTR_FETCH_DUP_RATIO` logical name, the `BTR_FETCH_DUP_RATIO` configuration variable, or the “B-tree duplicate fetch threshold” option from the configuration submenu.

m>% duplicate b-tree store above <n.m>% threshold

This message indicates that the average number of b-tree index duplicate node stores exceeds the corresponding threshold.

The default b-tree index duplicate store threshold is 15%. You can override the default value with the `RDMSBIND_STATS_BTR_STORE_DUP_RATIO` logical name, the `BTR_STORE_DUP_RATIO` configuration variable, or the “B-tree duplicate store threshold” option from the configuration submenu.

m>% leaf-node b-tree fetch below <n.m>% threshold

This message indicates that the average number of b-tree index leaf-node retrievals is less than the corresponding threshold.

This may be an indication that the b-tree index contains too many levels. You may want to adjust the fan-out factors during index creation to use a larger b-tree that contains fewer levels. This may help improve overall b-tree index throughput and utilization.

The default lock stall threshold is 25%. You can override the default value with the `RDMSBIND_STATS_BTR_LEF_FETCH_RATIO` logical name, the `BTR_LEF_FETCH_RATIO` configuration variable, or the “B-tree leaf node store threshold” option from the configuration submenu.

m>% duplicate hash index fetch above <n.m>% threshold³⁰

This message indicates that the average number of hash index duplicate node fetches exceeds the corresponding threshold.

The default hash index duplicate fetch threshold is 15%. You can override the default value with the RDMSBIND_STATS_HASH_FETCH_DUP_RATIO logical name, the HASH_FETCH_DUP_RATIO configuration variable, or the “Hash duplicate fetch threshold” option from the configuration submenu.

m>% duplicate hash index store above <n.m>% threshold³¹

This message indicates that the average number of hash index duplicate node stores exceeds the corresponding threshold.

The default hash index duplicate store threshold is 15%. You can override the default value with the RDMSBIND_STATS_HASH_STORE_DUP_RATIO logical name, the HASH_STORE_DUP_RATIO configuration variable, or the “Hash duplicate store threshold” option from the configuration submenu.

³⁰ Oracle Rdb 7.0.2.0


³¹ Oracle Rdb 7.0.2.0

ROW CACHE
ANALYSIS SCREEN

The “Row Cache Analysis” screen provides information about row cache effectiveness and performance.

Row cache is not enabled This message indicates that the row caching feature is not available on your database. There is no threshold associated with this message.


The following requirements must be met in order for the row caching feature to be enabled:


 At least 1 storage area must be enabled for record caching


Note that logical area row caches are not considered when determining whether or not row caching is enabled.

Row cache is not allowed This message indicates that the row caching feature is not allowed on your database. There is no threshold associated with this message.

The following requirements must be met in order for the row caching feature to be allowed:

 Row Caching must be enabled (see above).

 At least one row cache must be active.

 The database node count must be “1”.

 The Fast Commit feature must be enabled.

The Fast Commit feature requires the following:

 After-image journaling must be enabled.

 Fast commit must be enabled.

 All snapshot files must be allowed.

<rcache_name> Hash Queues: This message indicates that the hash queues for the identified row cache exceed the corresponding threshold.

The default lock stall threshold is 25%. You can override the default value with the RDM\$BIND_STATS_MAX_HASH_QUE_LEN logical name, the MAX_HASH_QUE_LEN configuration variable, or the “Hash table queue length threshold” option from the configuration submenu.

P R O C E S S
 A N A L Y S I S
 S C R E E N S

The “Process Analysis” screen provides information about process operation and performance. You can use the information in this screen to examine the effectiveness of virtual memory (VM), I/O operations and locking operations.

This screen is only available on OpenVMS.

Process n ENQLM <x.y> ratio below <x.y>% threshold

This message indicates that the process ENQCNT to ENQLM ratio is below the corresponding threshold.

The default ENQLM threshold is 25%. You can override the default value with the RDM\$BIND_STATS_PROC_ENQLM_RATIO logical name, the PROC_ENQLM_RATIO configuration variable, or the “ENQLM threshold” option from the configuration submenu.

Process n ASTLM <x.y> ratio below <x.y>% threshold

This message indicates that the process ASTCNT to ASTLM ratio is below the corresponding threshold.

The default BIOLM threshold is 25%. You can override the default value with the RDM\$BIND_STATS_PROC_ASTLM_RATIO logical name, the PROC_ASTLM_RATIO configuration variable, or the “ASTLM threshold” option from the configuration submenu.

Process n BIOLM <x.y> ratio below <x.y>% threshold

This message indicates that the process BIOCNT to BIOLM ratio is below the corresponding threshold.

The default BIOLM threshold is 25%. You can override the default value with the RDM\$BIND_STATS_PROC_BIOLM_RATIO logical name, the PROC_BIOLM_RATIO con-

figuration variable, or the “BIOLM threshold“ option from the configuration sub-menu.

Process n DIOLM <x.y> ratio below <x.y>% threshold

This message indicates that the process DIOCNT to DIOLM ratio is below the corresponding threshold.

The default DIOLM threshold is 25%. You can override the default value with the RDM\$BIND_STATS_PROC_DIOLM_RATIO logical name, the PROC_DIOLM_RATIO configuration variable, or the “DIOLM threshold“ option from the configuration sub-menu.

Process n PGFLQUOTA <x.y> ratio below <x.y>% threshold

This message indicates that the process PAGFILCNT to PGFLQUOTA ratio is below the corresponding threshold.

The default PGFLQUOTA threshold is 25%. You can override the default value with the RDM\$BIND_STATS_PROC_PGFLM_RATIO logical name, the PROC_PGFLM_RATIO configuration variable, or the “PGFLQUOTA threshold“ option from the configuration submenu.

S U M M A R Y

The “Online Analysis” facility is an extremely powerful tool for the integrated analysis of realtime performance characteristics of your database. The facility is not designed to replace or supplement the Oracle Trace product, which performs much more detailed analysis. However, the “Online Analysis” facility does perform a reasonable job of analyzing specific statistics characteristics and interactions with other statistics, in a most timely manner.

Diagnosing Performance Problems

Chapter 11 - Problem Diagnosis

INTRODUCTION

A database application is normally I/O intensive, so this chapter suggests that you analyze resources in the following order:

1. Process Information
2. I/O Information
3. Locking Information

Analyzing Process Information

STALL MESSAGE
ANALYSIS

The “Stall Messages” screen shows a summary of database users that are actively stalled for a database resource. Users stall on a resource whenever Oracle Rdb issues a synchronous database operation on behalf of the user's process. For example, a stall occurs when a user waits for a lock request or for completion of a physical disk read or write. A stall can also include operations such as AIJ switch-over, which does not involve any operating system interaction.

By default, the “Stall Messages” screen shows all stalls, including those of millisecond duration. When a high-performance, high-volume online transaction processing (OLTP) application issues a large number of I/O operations on a high-speed disk device, a database administrator (DBA) may find it impossible to differentiate between the many short millisecond stalls of the OLTP application and the longer, more important stalls that may be encountered by other applications using the database.

By typing **A** to select the "Alarm" on-screen menu option from the "Stall Messages" screen, you can specify a duration, in whole seconds, that a process must stall before it appears on the "Stall Messages" screen. For example, if you specify an alarm interval of "5" seconds, then only stalls of 5 seconds or longer duration will appear on the "Stall Messages" screen. If you specify the value "0" as the alarm interval, which is the default value, all active stalls will appear on the "Stall Messages" screen.

By entering **B** once to select the "Bell" on-screen menu option, you can activate the alarm bell option and the option will be highlighted. Entering **B** again will deactivate the alarm bell and the option will not be highlighted.

The alarm bell, even if activated, will ring only if the "Alarm" on-screen menu option is active.

When both the alarm and the alarm bell are active, the alarm bell will sound once per screen refresh, specified by the "Set_rate" on-screen menu option, if there are any displayed stalls. The alarm is deactivated by setting the alarm interval to zero seconds.

The following example shows a "Stall Messages" screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:29:32.26
Rate: 0.50 Seconds           Stall Messages           Elapsed: 00:02:34.71
Page: 1 of 1       DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1       Mode: Online
-----
```

```
Process.ID Since..... T Stall.reason..... Lock.ID.
3800C2B5:1 10:29:31.63 W waiting for page 1:16 (PW)           1E0066B0
3800C2B8:1 10:29:31.78 W waiting for page 30:47 (PW)          3E00B3E0
38009EB7:1 10:29:32.30 W waiting for async-write of pages 1:26 to 1:26
3800BCB9:1s10:29:32.32 - hibernating on AIJ I/O completion
3800BCB6:1 10:29:32.32 W hibernating on AIJ submission
```

```
-----
Alarm Bell Config Exit Filter Help LockID Menu >next_page <prev_page PageInfo Se
```

This screen lists database user processes and describes the most recent stalls executed by users on the node from which the RMU Show Statistic utility was invoked. Note that most stalls are extremely short-duration and consequently are not reported, since the stall messages are sampled only at the screen update interval. If the same stall message for a process persists, it could indicate a problem. The display also shows when a process is writing a bugcheck dump; a bugcheck dump file name longer than 53 characters is truncated.

The "Stall Messages" screen shows only processes that are currently stalling. Once a process finishes stalling, it disappears from the screen. Processes that are still stalling ripple up to the top of the screen. This means that the longest stalling processes appear at the top of the screen. Newer stalls are added to the bottom of the screen. Therefore, all users on the same node share the same stall display lines, and only the actively stalling processes show up on the stall screen. This allows you to monitor a relatively large number of stalling processes.

A database with no stalls has a blank stall display.

If more stalls are in progress than can fit on your screen, some current stalls might not be displayed. The RMU Show Statistic utility attempts to place as many current stall messages on the screen as possible. However, since the stalls are sorted in descending stall duration, the first page of the “Stall Messages” screen always contains the most important stalls to be analyzed.

You can redirect stall messages to a file by using the STALL_LOG qualifier to the RMU Show Statistics command and specifying a file name. You can also start and stop the logging of stall messages to a file from within the RMU Show Statistic utility at runtime. Invoke the Tools menu by entering the exclamation point (!), and select the “Start/Stop stall logging” menu option.

Look for prolonged or common stalls. Do a certain set of pages appear regularly? Perhaps a sorted index with duplicates is showing excessive pages being read.

INTEGRATED PAGE
 INFORMATION
 DISPLAY

The RMU Show Statistic utility provides integrated data page information on the “Stall Messages”, “Active User Stall Messages”, and “DBR Activity” screens.

Type “**P**” to display a menu of any DBKEYs currently displayed on the active screen.

For example, consider the following “Active User Stall Messages” screen:

```

Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 10-FEB-1998 10:36:45.21
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 00:01:09.58
Page: 1 of 1          DISK$: [DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1    Mode: Online
-----
Process.ID Since..... T Stall.reason..... Lock.ID.
3800090B:1          locking page 32:3
38019D0D:1 10:36:39.23 W waiting for record 50:60:0 (PR)          4600259F
-----
Config Exit Help LockID Menu >next_page <prev_page PageInfo Set_rate Write Zoom
  
```

The process 38019D0D:1 has been waiting for the record 50:60:0 for several seconds. Typing "P" displays the menu of processes with DBKEYs in their stall message. For example:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 10-FEB-1998 10:36:45.21
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 00:01:09.58
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
Process.ID Since..... T Stall.reason..... Lock.ID.
A. 3800090B:1          locking page 32:3
B. 38019D0D:1 36:39.23 W waiting for record 50:60:0 (PR)      4600259F
```

```
-----
Config Exit Help LockID Menu >next_page <prev_page PageInfo Set_rate Write Zoom
```

Note that both processes are in the menu, even though only one of the processes is actually stalled. Selecting the stalled process displays the following page information:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 10-FEB-1998 10:36:45.21
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 00:01:09.58
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
Process.ID Since..... T Stall.reason..... Lock.ID.
3800090B:1          locking page 32:3
38019D0D:1 10:36:39.23 W waiting for record 50:60:0 (PR)      4600259F
```

```
+-----+
      0003 0000003C 0000 page 60, physical area 3 (live)
          15B70BF1 0006 checksum = 15B70BF1
009C1998 3B938625 000A time stamp = 10-FEB-1998 10:36:39.
          0000 0366 0012 870 free bytes, 0 locked
              0004 0016 4 lines
          03E2 000B 0018 line 0: offset 03E2, 11 bytes
          03D0 0011 001C line 1: offset 03D0, 17 bytes
          03B0 0020 0020 line 2: offset 03B0, 32 bytes
          039E 0011 0024 line 3: offset 039E, 17 bytes
          00000120 0028 line 0: TSN 288
          00000120 002C line 1: TSN 288
          00000120 0030 line 2: TSN 288
          00000121 0034 line 3: TSN 289
          00000001 03EE snap page pointer 1
          00000121 03F2 snap pointer TSN 289
          00000004 03F8 page sequence number 4
              0000 03FC page TSN base 0
+-----+
```

```
-----
Config Exit Help LockID Menu >next_page <prev_page PageInfo Set_rate Write Zoom
```

The "Page Information" zoom screen indicates that line 0 is probably a "system record" for hash indexes. In this example, the TSN 0:288 has probably not committed.

Further analysis may require use of the RMU Dump utility to display the actual line contents, in order to confirm any hypothesis's derived using the RMU Show Statistic utility.

STALL MESSAGE
LOCKING
ANALYSIS

The RMU Show Statistic utility provides the following screens to help gather information about locking:

- The “Stall Messages” and “Active User Stall Messages” screens identify stalls for lock requests.
- The “Summary Locking Statistics” screen shows a summary of the lock activity done by Oracle Rdb.
- The “Locking (One Lock Type)” screen shows lock statistics for a particular lock type. You should use this screen when you want all the available information on a specific lock type. The only difference between this lock screen and the “Locking (One Stat Field)” screen is that these statistics pertain only to the type of lock that you specify and are, therefore, more useful for detailed analysis of lock activity.
- The “Locking (One Stat Field)” screen shows lock statistics for all the lock types of a particular statistical field. You should use this screen when you want to see specific information about all the different lock types.
- The “Lock Deadlock History” screen can be used to identify the object that causes a deadlock event.
- The “Lock Timeout History” screen can be used to identify the object that causes a timeout event.
- The “Lock Statistics (by file)” screen displays information about page locks that are specific to storage areas and snapshot files. This information is vital in determining which storage areas have the most locking activity, and analyzing the validity of storage area partitioning.

INTEGRATED LOCK
INFORMATION
DISPLAY

The RMU Show Statistic utility provides integrated lock information on the “Stall Messages”, “Active User Stall Messages”, and “DBR Activity” screens.

Type “**L**” to display a menu of any lock identifiers currently displayed on the active screen.

For example, consider the following RMU Show Statistic utility “Active User Stall Messages” screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:29:32.26
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 03:06:07.20
Page: 1 of 1          KODD$:[DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

```
Process.ID Since.....  Stall.reason.....  Lock.ID.2B600554:1
15:38:27.04 - waiting for record 55:321:2 (EX)      1500624B2B600556:1
15:40:40.66 - waiting for record 55:321:2 (EX)      1200A2E42B600555:1
writing pages back to database
```

```
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

Typing “L” displays the menu of available lock identifiers. For example:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:29:32.26
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 03:06:07.20
Page: 1 of 1          KODD$:[DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

```
Process.ID Since.....  Stall.reason.....  Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX)      A. 1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX)      B. 1200A2E4
2B600555:1      writing pages back to database
```

```
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

Note the displayed lock identifiers have been prefixed with the standard menu-selection letter choice. The desired lock identifier can be selected by either typing the desired letter, or moving the highlighted cursor using the up arrow (↑) or down arrow (↓) keys and then pressing the Return key.

If there are no lock identifiers on the display, typing “L” has no effect.

When the desired lock identifier is selected, the “Lock Information” zoom screen appears. For example:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:29:32.26
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 03:06:07.20
Page: 1 of 1          KODD$: [DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1      Mode: Online
```

```
-----
Process.ID Since.....  Stall.reason.....  Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX)      >1500624B<
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX)      1200A2E4
2B600555:1                writing pages back to database
```

```
+-- Lock Information: 1500624B -----+
|                                     |
|           Resource: record 321:2   |
| State... ProcessID Process.Name... Lock.ID.  Rq Gr Queue |
|-----|-----|-----|-----|-----|-----|
| Blocker: 2B600555  RICK3.....  0F005AB6  EX EX Grant |
| Waiting: 2B600556  RICK4.....  1200A2E4  EX NL Cnvrt |
| Waiting: 2B600554  RICK2.....  1500624B  EX NL Cnvrt |
|-----|-----|-----|-----|-----|-----|
+-----+

```

```
-----
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

The “Lock Information” zoom screen is not dynamic. The lock information is captured at the time the lock identifier is selected, and the information does not dynamically change. The “Lock Information” display is a snapshot in time of the selected lock identifier.

Of course, the “Lock Information” zoom screen can be saved to disk using the “Write” on-screen menu option.

Note that while the order of displayed locks is arbitrary, you can identify the order in which waiting locks will be granted. For instance, in the previous example, two processes are waiting for the same lock. Displaying the information for lock 1500624B simply indicates that both processes are waiting for the same resource.

Displaying the information for lock 1200A2E4 shows the following information:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:29:32.26
Rate: 1.00 Second      Active User Stall Messages      Elapsed: 03:06:07.20
Page: 1 of 1          KODD$: [DB_HOMEDIR.WORK.AIJ]MF_PERSONNEL.RDB;1      Mode: Online
```

```
-----
Process.ID Since.....  Stall.reason.....  Lock.ID.
2B600554:1 15:38:27.04 - waiting for record 55:321:2 (EX)      1500624B
2B600556:1 15:40:40.66 - waiting for record 55:321:2 (EX)      1200A2E4
2B600555:1                               writing pages back to database
```

```
+-- Lock Information: 1500624B -----+
|                                     |
|           Resource: record 321:2   |
| State... ProcessID Process.Name... Lock.ID.  Rq Gr Queue |
|-----|-----|-----|-----|-----|-----|
| Blocker: 2B600555  RICK3.....  0F005AB6  EX EX Grant |
| Blocker: 2B600554  RICK2.....  1500624B  EX NL Cnvrt |
| Waiting: 2B600556  RICK4.....  1200A2E4  EX NL Cnvrt |
|-----|-----|-----|-----|-----|-----|
+-----+
-----
```

```
Config Exit Help LockID Menu >next_page <prev_page Set_rate Write Zoom !
```

This display indicates that, although lock 1200A2E4 is on the conversion queue, it is blocking the other lock and will be granted before the other lock.

Because the database is continuing to operate while you display the lock information, the selected lock can be released before its information can be displayed. If this happens, the message “Lock has been released since it was displayed” is displayed on the terminal screen. Also, the lock identifier menu is canceled when you return from the “Lock Information” screen.

If you do not have sufficient privilege to retrieve the lock information, the message “Insufficient privilege to display lock information” is displayed on the terminal screen. Also, the lock identifier menu is canceled when you return from the “Lock Information” zoom screen.

Because of the dynamic nature of the lock information, the lock information is not written to the output file and cannot be replayed from an input file.

While the “Lock Information” zoom screen is displayed, terminal broadcast messages are suspended. The messages resume after you leave the “Lock Information” display.



Note that on the “Active User Stall Messages” screen, stalled locks that are no longer stalled are still displayed. Even if the displayed lock is no longer stalled, you can still attempt to display information on that stalled lock. However, the lock that was stalled is probably released.

The “Lock Information” zoom screen does not attempt to provide detailed information about the process holding a respective lock. The process information can be obtained using the “Process Accounting” screen if the process is on the current node.

Because of the current non-splitting menu limitation of 36 options, using the menu-selection letters A to Z and 0 to 9, only the first 36 lock identifiers on a given display are selectable. This restriction may be removed in the future. To work around this restriction, set the terminal screen length to less than 42 lines or use the other “Stall Messages” screen, where the order of the displayed lock identifiers may be different from the current screen.

LOCK DEADLOCK
HISTORY SCREEN

The “Lock Deadlock History” screen can be used to identify the object that causes a deadlock event. You access the “Lock Deadlock History” screen from the “Process Information” submenu.

The “Stall Messages” screen provides stall information, but when a lock is deadlocked, the stall is terminated and the information is no longer available on the “Stall Messages” screen.

For each active process on the current node, the “Lock Deadlock History” screen shows the process identifier, the time that the process was most recently involved in a deadlock, the reason for the most recent deadlock, and the number of deadlocks the process has encountered since attaching to the database.

The following “Lock Deadlock History” screen shows a record deadlock. The lock deadlock reason indicates that the process was waiting for a record and provides the DBKEY of that record.

By using the “Lock Deadlock History” screen to examine the most recent deadlocks, a DBA can more easily identify potential database hot spots and application bottlenecks. Looking at a screenful of deadlock information can help the DBA differentiate between frequent and infrequent problems and correlate common causes of the problems.

In general, record deadlocks are undesirable and the cause of such deadlocks should be identified and fixed immediately. Page deadlocks are not always bad, but they need to be analyzed, as all deadlocks cause application throughput to degrade.

LOCK TIMEOUT
HISTORY SCREEN

The “Lock Timeout History” screen can be used to identify the object that causes a timeout event. You access the “Lock Timeout History” screen from the Process Information submenu.

The “Stall Messages” screen provides stall information, but when a process times out while waiting for a lock, the stall is terminated and the information is no longer available on the “Stall Messages” screen.

For each active process on the current node, the “Lock Timeout History” screen shows the process identifier, the time that the process most recently timed out while waiting for a lock, the reason for the most recent timeout experienced by the process, and the number of timeouts experienced by the process since attaching to the database.

PROCESS
 ACCOUNTING
 SCREEN

The “Process Accounting” screen provides continuously updated accounting information about local processes. This screen is an alternative to the OpenVMS SHOW PROCESS/CONTINUOUS utility. The screen provides equivalent information, except that all active database processes on a specific node can be monitored at the same time.

This screen shows direct operating system accounting information, thereby enabling a database administrator to evaluate the system resources used by database processes. The values in the “Process Accounting” screen are for all process activity, not just the activity that occurs while in the database. Therefore, this screen is useful for monitoring the complete application behavior.

You access the “Process Accounting” screen from the “Process Information” submenu.

This screen shows dynamically changing process information only. That is, quotas and other information that are fixed for each process are not displayed because that information can be obtained in other ways. The “Process Accounting” screen has brief and full modes that control the amount of information displayed for each active database process.

You select brief mode by typing “**B**”. In brief mode, one line per process is displayed, providing the following information: process identifier, process name, CPU time, remaining lock quota count, page fault count, number of direct I/O operations, working set size, and virtual memory size. You select full mode by typing “**F**”. In full mode, a second line per process is displayed that provides the following additional information: user name, image name, process state, page file quota count, direct I/O quota count, number of buffered I/O operations, and buffered I/O quota count.

The following is an example of the "Process Accounting" screen in full mode:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:37:46.90
Rate: 3.00 Seconds      Process Accounting              Elapsed: 00:22:39.16
Page: 1 of 1           SQL_DISK1:[RICK.V70]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

Process.ID	Process.name...	CPUtime....	EnqCnt.	PGflts.	NumDio.	WSsize.	VMsize.
User.name...	Image.name.....	State..	PGfCnt.	DioCnt.	NumBio.	BioCnt.	
6E2012D3:1	RICK	00:00:15.55	17796	20776	808	4472	15082
RICK	SQL\$601		LEF	124448	200	1464	199
6E201887:1	SMITH	00:00:11.92	17858	18378	516	5000	14954
SMITH	SQL\$601		LEF	124576	200	515	199

```
-----
Brief Exit Help Menu >next_page <prev_page Set_rate Write Zoom !
```

Note that information on the "Process Accounting" screen cannot be reset. Because the information is gathered in real time, it cannot be written to the output file (the OUTPUT qualifier will not work with this screen) and, therefore, cannot be displayed during input file replay.

CHECKPOINT
INFORMATION
SCREEN

The "Checkpoint Information" screen displays process checkpoint information. One line of information is displayed for each process attached to the database on the current node. There may be blank lines in the display as processes detach from the database; in this way, once a process has attached to the database, it maintains the same location in the screen until it detaches from the database.

The following is an example of a "Checkpoint Information" screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 14:51:50.47
Rate: 1.00 Second      Checkpoint Information (Unsorted)   Elapsed: 00:01:15.29
Page: 1 of 1           DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;2      Mode: Online
-----
```

Process.ID	Ckpt.Vno:	Ckpt.Vbn	QuietVno	ST	CurTSN	Tx.Start.Time	AIJ: 1:558
38004030:1	1	12		1	RW 0:1296	14:51:50.10	TSN: 0:1294
38004036:1	1	11		1	RW 0:1298		
38003245:1s							
3800AC37:1	1	13		1	RW 0:1294	14:51:50.04	
38006444:1	1	14		1	RW 0:1297	14:51:50.19	

```
-----
Alarm Bell Config Exit Help Menu >next_page <prev_page Refresh Set_rate Write Zo
```

Analyzing I/O Resources

DATABASE
PARAMETER
INFORMATION
SCREENS

The RMU Show Statistic utility provides a set of screens that display dynamic information. The information on these screens automatically changes to reflect database parameter modifications. The “Database Parameter Info” submenu contains a variety of screens depicting various modifiable database information, in a format similar to the RMU Dump Header utility.

The following example shows a “Buffer Information” screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor   3-FEB-1998 05:39:34.73
Rate: 1.00 Second      Buffer Information              Elapsed: 00:00:16.33
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
```

```
Default user buffer count is 20
Default recovery buffer count is 20
Buffer size is 6 blocks
Maximum pages per buffer is 3
Global Buffers are disabled
- Global buffer count is 250
- Maximum global buffer count per user is 5
- Page transfer via memory is disabled
Global section size with global buffers disabled is 1131069 bytes
- With global buffers enabled is 2061787 bytes
Asynchronous pre-fetch is enabled
- Maximum pre-fetch depth is 5 buffers
Detected asynchronous pre-fetch is enabled
- Maximum pre-fetch depth is 4 buffers
- Pre-fetch threshold is 4 pages
Asynchronous batch-write is enabled
- Clean buffer count is 5
- Maximum batch size is 4 buffers
-----
```

```
Exit Help Menu Options Refresh Set_rate Write !
```

The “Database Parameter Info” screens are not recorded in the binary output file produced using the OUTPUT qualifier. Consequently, these screens are not available when you replay a binary file using the INPUT qualifier. However, you can use the “Write” on-screen menu option, or the “Options” on-screen menu option to save the screens to a disk-based file.

PIO STATISTICS--
DATA WRITES
SCREEN

The “PIO Statistics--Data Writes” screen provides information concerning data file writes and buffer unmarking activity (buffer writes to the disk).

The following is an example of a "PIO Statistics--Data Writes" screen:

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:07.61
 Rate: 1.00 Second PIO Statistics--Data Writes Elapsed: 00:40:01.05
 Page: 1 of 1 DISK\$: [DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....
	max.....	cur.....	avg.....		
unmark buffer	66	11	29.8	71595	3.9
transaction	29	0	0.0	31	0.0
pool overflow	63	0	23.2	55839	3.0
blocking AST	0	0	0.0	0	0.0
lock quota	0	0	0.0	0	0.0
lock conflict	53	11	6.2	14916	0.8
user unbind	2	0	0.0	3	0.0
batch rollback	0	0	0.0	0	0.0
new area mode	0	0	0.0	0	0.0
larea change	7	0	0.0	31	0.0
incr backup	0	0	0.0	0	0.0
no AIJ access	0	0	0.0	0	0.0
truncate snaps	0	0	0.0	0	0.0
checkpoint	44	0	0.3	777	0.0
AIJ backup	0	0	0.0	0	0.0

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

PIO STATISTICS--
 DATA FETCHES
 SCREEN

The "PIO Statistics--Data Fetches" screen provides statistics on how data page requests are handled. There are two different formats for this screen: one for databases that have local buffers enabled, and another for databases that have global buffers enabled.

The following is an example of a "PIO Statistics--Data Fetches" screen for a database with local buffers enabled:

```

Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:01.21
Rate: 1.00 Second      PIO Statistics--Data Fetches      Elapsed: 00:39:54.65
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
-----
statistic.....      rate.per.second..... total..... ave rage.....
name.....           max..... cur..... avg..... count..... per.trans....

fetch for read          1033      125      140.5      336503      18.5
fetch for write         1042      324      424.1      1015673      56.0

in LB: all ok           1307      301      477.3      1143037      63.0
  LB: need lock          65         8       15.1      36331       2.0
  LB: old version        66         5        2.4       5809        0.3

not found: read         290      134      69.4      166366      9.1
      : synth            12         0        0.1        342         0.0

DAPF: success           0         0        0.0         0         0.0
DAPF: failure           0         0        0.0         0         0.0
DAPF: utilized          0         0        0.0         0         0.0
DAPF: discarded         0         0        0.0         0         0.0
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !
  
```

The following is an example of a "PIO Statistics--Data Fetches" screen for a database with global buffers enabled:

```

Node: MYNODE (1/1/2)   Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:35:33.27
Rate: 1.00 Second      PIO Statistics--Data Fetches      Elapsed: 00:04:10.48
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
-----
statistic.....      rate.per.second..... total..... ave rage.....
name.....           max..... cur..... avg..... count..... per.trans....

fetch for read          1285      220      53.6      13449      15.1
fetch for write          522      431      87.7      21986      24.8
in AS: all ok           921      611      126.0      31580      35.6
  AS: lock for GB         0         0        0.0         0         0.0
  AS: need lock           181        29      10.6      2662       3.0
  AS: old version         0         0        0.0         0         0.0
in GB: need lock        176        11       4.2      1064       1.2
  GB: old version         0         0        0.0         0         0.0
  GB: transferred         0         0        0.0         0         0.0
not found: read         24         0        0.5      126        0.1
      : synth             2         0        0.0        13         0.0

DAPF: success           0         0        0.0         0         0.0
DAPF: failure           0         0        0.0         0         0.0
DAPF: utilized          0         0        0.0         0         0.0
DAPF: discarded         0         0        0.0         0         0.0
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !
  
```

PIO STATISTICS--
 SPAM FETCHES
 SCREEN

The "PIO Statistics--SPAM Fetches" screen provides statistics on how SPAM page requests are handled. There are two different formats for this screen--one for databases that have local buffers enabled, and another for databases that have global buffers enabled.

The following is an example of a "PIO Statistics--SPAM Fetches" screen for a database with local buffers enabled:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:04.36
Rate: 1.00 Second      PIO Statistics--SPAM Fetches      Elapsed: 00:39:57.80
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
```

```
-----
```

statistic.....	rate.per.second.....			total.....	ave rage.....
name.....	max.....	cur.....	avg.....	count.....	per.trans....
fetch for read	537	28	20.8	50020	2.7
fetch for write	29	0	3.0	7274	0.4
in LB: all ok	562	9	15.5	37296	2.0
LB: need lock	11	1	1.4	3368	0.1
LB: old version	5	0	0.2	603	0.0
not found: read	45	18	6.6	16030	0.8
: synth	0	0	0.0	0	0.0
DAPF: success	0	0	0.0	0	0.0
DAPF: failure	0	0	0.0	0	0.0
DAPF: utilized	0	0	0.0	0	0.0
DAPF: discarded	0	0	0.0	0	0.0

```
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !
```

ASYNCHRONOUS
 IO STATISTICS
 SCREEN

The "Asynchronous IO Statistics" screen provides information concerning asynchronous reads and writes to the database files. The stall times are displayed in hundredths of a second.

This screen shows the number of requests for asynchronous read and write operations, as well as the number of asynchronous read and write operations actually done.

You access the "Asynchronous IO Statistics" screen from the "IO Statistics Information" submenu.

The following is an example of the "Asynchronous IO Statistics" screen:

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:14.20
 Rate: 1.00 Second Asynchronous IO Statistics Elapsed: 00:40:07.64
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....	
	max.....	cur.....	avg.....			
data read request	0	0	0.0	0	0.0	0.0
data read IO	0	0	0.0	0	0.0	0.0
spam read request	0	0	0.0	0	0.0	0.0
spam read IO	0	0	0.0	0	0.0	0.0
read stall count	0	0	0.0	0	0.0	0.0
read stall time	0	0	0.0	0	0.0	0.0
write IO	42	16	14.4	34700	1.9	1.9
write stall count	19	6	4.4	10616	0.5	0.5
write stall time	93	31	18.0	43545	2.3	2.3

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

RECORD
 STATISTICS
 SCREEN

The "Record Statistics" screen shows a summary of data row activity (rows modified, fetched, stored, or erased) for storage areas in the database.

The following is an example of a "Record Statistics" screen:

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 23-JAN-1998 14:30:52.84
 Rate: 1.00 Second Record Statistics Elapsed: 00:58:16.96
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....	
	max.....	cur.....	avg.....			
record marked	0	0	0.0	0	0.0	0.0
record fetched	0	0	0.0	0	0.0	0.0
fragmented	0	0	0.0	0	0.0	0.0
record stored	0	0	0.0	0	0.0	0.0
fragmented	0	0	0.0	0	0.0	0.0
pages checked	0	0	0.0	0	0.0	0.0
saved IO	0	0	0.0	0	0.0	0.0
discarded	0	0	0.0	0	0.0	0.0
record erased	0	0	0.0	0	0.0	0.0
fragmented	0	0	0.0	0	0.0	0.0

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot

SNAPSHOT
 STATISTICS
 SCREEN

The "Snapshot Statistics" screen shows snapshot activity for both update and read-only transactions.

The following is an example of a "Snapshot Statistics" screen:

```
Node: MYNODE (1/1/1)      Oracle Rdb X7.0-00 Perf. Monitor 23-JAN-1998 15:30:49.46
Rate: 1.00 Second          Snapshot Statistics              Elapsed: 00:24:14.88
Page: 1 of 1      DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....	
	max.....	cur.....	avg.....			
Total transactions	0	0	0.0	0	0.0	0.0
R/O transactions	0	0	0.0	0	0.0	0.0
retrieved record	0	0	0.0	0	0.0	0.0
fetched line	0	0	0.0	0	0.0	0.0
read snap page	0	0	0.0	0	0.0	0.0
stored snap record	0	0	0.0	0	0.0	0.0
page in use	0	0	0.0	0	0.0	0.0
page too full	0	0	0.0	0	0.0	0.0
page conflict	0	0	0.0	0	0.0	0.0
extended file	0	0	0.0	0	0.0	0.0

```
-----
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !
```

DETECTING I/O
 RESOURCE
 BOTTLENECKS

When you review I/O resources, you should concentrate on detecting I/O resource bottlenecks, balancing I/O load, and reducing I/O operations.

The best place to start investigating your system for a potential I/O resource bottleneck is at the disk devices. If the utilization is over 65 percent, performance is likely to be affected to some extent.

If you have a single-file database and performance is a bigger concern than reduced database complexity, your first step is probably to change it to a multifile database. You can determine the approximate maximum throughput for a transaction in a single-file database to see whether it is worthwhile to change the database to a multifile database. The formula for determining the maximum throughput for a given transaction in a single-file database is the following:

$$\text{Transactions per Second} = \frac{\text{Disk I/Os Performed per Second}}{(\# \text{ of data I/Os}) + (\# \text{ of control I/Os})}$$

To determine the maximum possible throughput for a transaction in your single-file database, you first need to know approximately how many I/Os per second can be performed by the database disks.

The previous calculation assumes that the disk devices can perform 30 I/Os per second. Second, you need to know the number of data I/Os to write to the storage area files (.RDA) and snapshot files (.SNP) files that must be performed by the transaction (a value of “7” in our example) and the number of I/Os to write to the database root-file (.RDB), recovery-unit journal (.RUJ), and after-image journals (a value of “3” in our example) for the transaction.

So, for our example, the maximum possible throughput for the transaction is 3 transactions per seconds (TPS), as shown in the following:

$$3 \text{ Transactions per Second} = \frac{30}{7 + 3}$$

If 7 I/O operations are performed by the transaction, the best throughput you can expect in a single-file database is 3 transactions per second (TPS). If you want to improve the throughput, you should change the database to a multifile database. Note that just making the database multifile does not yield large performance gains. Rather, the multifile option gives you the potential to explore more options during the tuning process. If a bottleneck develops in a multifile database, you still must analyze and understand where the bottleneck is, and make the appropriate changes to remove it.

If a database is a multifile database, one tuning option is to spread it across multiple disks to reduce I/O contention on the single device. If you find that one of your database disks is still highly utilized, you have several options. If additional disks are available, spread the database out more to take advantage of them. If this is not an option, then you need to improve the balance of the I/O load across the existing disk resources. When you spread or rebalance your database, remember that the database root and after-image journals can become database hot spots in a high transaction environment. The root file hot spot can be alleviated by using the Fast Commit feature and the Commit To Journal Optimization feature, if applicable.

Also, the root file and after-image journals should each be on its own disk. If the root file cannot be placed on its own disk, place it on a disk that is not highly utilized. If you are operating in an environment that has different types of disk devices, you should place the root file and after-image journals on the fastest devices.

If the disk devices do not seem to be a problem, another area to check for I/O resource bottlenecks is the application. If an application has a high average direct I/O rate, it is possible that performance is being affected by the disk that services the requests. Because disk requests are relatively slow compared to memory requests, high direct I/O rates for an application are a potential area for tuning.

Once you have a multifile database, check to see that it is spread equally across the available disks, which can help reduce database hot spots. Also, if the default directo-

ries for database users are spread equally across the user disks, this can help prevent contention problems.

If the system disk is highly utilized, moving the default Oracle Rdb monitor log file location to another disk may help to reduce contention.

If the users' disks are highly utilized, you can define the logical names RDMSSBIND_SORT_WORKFILES, RDMSSRUJ, and RDMSSBIND_WORK_FILE to help reduce contention. These logical names deal with default placement of Oracle Rdb temporary files.

On OpenVMS, the RDMSSBIND_SORT_WORKFILES logical name defines the number of temporary files used by the OpenVMS Sort utility (SORT). Oracle Rdb uses the SORT utility for all sorting operations that the optimizer decides are too large to be done in memory, for example, large SORTED BY queries and large index builds. RDMSSBIND_SORT_WORKFILES has a maximum value of 10 files. The default value is 2.

While the RDMSSBIND_SORT_WORKFILES logical name controls the number of sort work files that are used, SORTWORK0 through SORTWORK9 control the placement of the sort work files. Distributing these files over a number of disks reduces contention during the sort operation.

You can also define the RDMSSBIND_SORT_WORKFILES logical name and its related sort work files for each application for finer granularity.

The RDMSSRUJ logical name controls the default placement of the recovery-unit journal (.RUJ) file. Defining this name is another way to reduce contention on a user disk. The following example shows how to redirect the .RUJ file on OpenVMS:

```
$ DEFINE RDMSSRUJ RUJ$DISK:[OFFLOADED_RUJS]
```

Finally, you can use the RDMSSBIND_WORK_FILE logical name to redirect temporary files created by Oracle Rdb to another location. You can also use this logical name to reduce contention on the user disk. The following example shows how to redirect the location where these temporary files are created on OpenVMS:

```
$ DEFINE RDMSSBIND_WORK_FILE -  
_$_WORK$DISK:[OFFLOADED_TEMP_FILES]
```

If the system disk remains a bottleneck after all the database files and repository files have been moved to other devices, it is possible that high page file activity is causing the bottleneck. If there is evidence of a high hard page fault rate and only a single page file is available for use, adding a secondary page file on another device may alleviate the bottleneck. This situation sometimes occurs in a database environment when process virtual memory requirements are much higher than their working set quotas. The database buffer parameters have the largest influence on process virtual memory needs.

CHECKING DATA
 DISTRIBUTION

One way to help determine I/O distribution across the database storage areas is to use the "IO Statistics screen" in the RMU Show Statistic utility. This gives you a general idea of how evenly distributed your I/O activity is across the storage areas and disks for a given time period. One way to measure this is to run the RMU Show Statistic utility for a period of time, review the "IO Statistics (by file)" screens, then bring up the "Option" on-screen menu and write a report. The "Option" on-screen menu option generates a file, "STATISTICS.RPT," that contains ASCII output of all the menus for that period.

You can edit the STATISTICS.RPT file using your favorite editor, or print it on any printer. In the STATISTICS.RPT file, you will find a section similar to the following:

```
Node: MYNODE (1/1/2)      Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 15:35:45.64
Rate: 1.00 Second       File IO Statistics           Elapsed: 02:04:22.85
Page: 1 of 1           DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
```

For File: Database Root

```

statistic..... rate.per.second..... total..... a verage.....
name..... max..... cur..... avg..... count..... per.trans....

total I/O rates                1                35845                1.1
  (Synch. reads)                0                0.0                26                0.0
  (Synch. writes)               5                0.8                6212               0.2
  (Extends)                    0                0.0                0                0.0
  (Asynch. reads)              0                0.0                0                0.0
  (Asynch. writes)            16                3.9                29607              0.9

statistic..... blocks.transferred..... stall.time.(x100).....
name..... avg.per.I/O.. total..... avg.per.I/O... total.....

total I/O sizes                1.0                36004                0.6                22477
  (Synch. reads)                1.8                48                0.3                9
  (Synch. writes)               1.0                6221                3.3                20914
  (Extends)                    0.0                0                0.0                0
  (Asynch. reads)              0.0                0                0.0                0
  (Asynch. writes)             1.0                29735                0.0                1554
-----
Exit Help Menu Options Reset Set_rate Write !

```

The "total I/O rates" section, namely the maximum and average per second, and the total count values, can be used to determine the percentage of I/Os that are going to each area. If the total I/O numbers are small, further optimization may yield only marginal results.

You also should look at the stall time values on the "IO Statistics" screen because long stall times cause the response time of an application to increase. When you are displaying statistics for all data files with the "IO Statistics" screen, as in the previous example, the value for the "stall time avg per I/O" field may not be what you expect if you have displayed the average stall time per I/O for the individual storage area files (.RDA) and snapshot files (.SNP) files in the database. For example, it is possible for the value in the "stall time avg per I/O" field for the all data files screen to be lower

than the average of the values for the “stall time avg per I/O” fields for the individual storage area files (.RDA) and snapshot files (.SNP) files.

Oracle Rdb issues write I/O operations in parallel, asynchronously (this is a batch-write mechanism). This means that the average stall time per I/O for the all data files screen is not the “average of the averages” shown for the individual storage area files (.RDA) and snapshot files (.SNP) files; this would imply that the I/Os completed serially. Rather, the total I/O stall time is “the average of the averages divided by the number of I/Os”; the average for the all data files screen is usually a fraction of the individual file averages because the stall time is amortized across all I/Os issued in parallel.

For example, assume that Oracle Rdb does a batch-write operation to four storage areas (all in parallel, of course). Assume that each individual storage area's I/O operation takes 20 milliseconds. If the I/Os were done serially, then the average stall time for all storage areas would be 20 milliseconds. However, because the I/Os are done in parallel, the average for all areas is actually 5 milliseconds (20 milliseconds divided by 4 I/O operations).

If asynchronous batch-write operations are enabled, Oracle Rdb performs write operations to the database, and does not have to wait for the write operation to complete. The I/Os for asynchronous writes are recorded on the “IO Statistics (by file)” screen, but no stall time is recorded on the “IO Statistics (by file)” screen for most of these writes because stalls are eliminated for most asynchronous batch-write operations.

If a disk takes 30 milliseconds for an I/O operation, then you should consider trying to improve (reduce) the stall times only when the average stall time per I/O on the “IO Statistics (by file)” screen is greater than 30 milliseconds. When the average stall time for an I/O operation is less than 30 milliseconds, this is attributable to the parallel writes performed by batch-write operations, as described earlier. You can examine disk performance statistics to monitor changes in disk performance.

REDUCING I/O
 OPERATIONS

One way to reduce the number of direct I/O operations is to prevent the need for them. This can be done by making changes at the system, database, or application level.

One way to control disk I/O is through buffer management. Oracle Rdb enables you to use local or global buffers.

The most frequently used way to reduce I/O from the system level is to make the buffer pool size bigger, so that the disk I/Os are replaced by extra memory references. To do this, additional memory capacity must exist on the system. Because access to the buffer is much faster than access to the disk, this approach can cause substantial improvements in response time, especially when the access is on either slow or highly utilized disk devices.

One way to control the I/O buffer pool size is to change the number of buffers that the database maintains. This can be done in the database using the NUMBER OF BUFFERS clause of the SQL ALTER DATABASE statement, or from the system level through the RDM\$BIND_BUFFERS logical name. Because this parameter is dynamic, it can be modified with very little risk or cost. The default number of database buffers is 20.

The logical name can be set system-, group-, or process-wide, for finer buffer granularity.

To help quantify the additional memory requirement due to increased numbers of database buffers, a database application was run in 20-buffer increments and memory statistics collected. Because buffer size remained constant, increasing the number of buffers affected virtual memory consumption linearly, as expected. Changing the buffer size should move the line a constant factor upwards or downwards, depending on in which direction buffer size is changed. However, results can differ dramatically depending on the exact nature of the application being studied; therefore, you will need to run similar experiments on your database to determine the additional amount of memory consumed by increasing the number of buffers.

What does this mean for response time? Response time will vary depending on disk speed and utilization, the type of transaction the application is performing, the frequency with which updates occur, and the amount of contention in the database. As a general rule, applications that perform range retrievals benefit from having more buffers. This is also true if the data is clustered, as associated information will be read into the buffers when the first piece of information is accessed. Load-intensive workloads benefit from having small numbers of buffers. Because load-intensive workloads deal primarily with the physical database page, it makes little sense for them to incur the overhead of maintaining a large number of buffers.

Response time is not linear with respect to the number of database buffers.

One tangible way to measure buffer pool effectiveness database-wide is through the “PIO Statistics--Data Fetches” and “PIO Statistics--SPAM Fetches” screens. The way that you determine your database’s buffer pool effectiveness depends on whether the database uses local buffers or global buffers.

For example:

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:01.21
 Rate: 1.00 Second PIO Statistics--Data Fetches Elapsed: 00:39:54.65
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

```

-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
fetch for read          1033      125      140.5      336503      18.5
fetch for write         1042      324      424.1      1015673      56.0

in LB: all ok           1307      301      477.3      1143037      63.0
  LB: need lock          65         8       15.1       36331        2.0
  LB: old version        66         5        2.4        5809         0.3

not found: read         290       134       69.4      166366        9.1
      : synth            12         0         0.1        342          0.0

DAPF: success           0         0         0.0         0            0.0
DAPF: failure           0         0         0.0         0            0.0
DAPF: utilized          0         0         0.0         0            0.0
DAPF: discarded         0         0         0.0         0            0.0
-----
  
```

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:24:04.36
 Rate: 1.00 Second PIO Statistics--SPAM Fetches Elapsed: 00:39:57.80
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

```

-----
statistic..... rate.per.second..... total..... ave rage.....
name..... max..... cur..... avg..... count..... per.trans....
fetch for read          537         28       20.8      50020         2.7
fetch for write          29          0         3.0      7274          0.4

in LB: all ok           562         9        15.5      37296         2.0
  LB: need lock          11          1         1.4       3368          0.1
  LB: old version         5           0         0.2         603          0.0

not found: read         45         18         6.6     16030         0.8
      : synth            0           0         0.0         0            0.0

DAPF: success           0         0         0.0         0            0.0
DAPF: failure           0         0         0.0         0            0.0
DAPF: utilized          0         0         0.0         0            0.0
DAPF: discarded         0         0         0.0         0            0.0
-----
  
```

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

For a database with local buffers enabled, you can calculate the percentage of page requests that find the page in the local buffer pool as follows:

$$(\text{total number found in local buffer pool} / \text{total number of page requests}) * 100$$

To calculate the “total number found in local buffer pool” count, add the sum of the “in LB: all ok” and “LB: need lock” counts on the “PIO Statistics--Data Fetches” screen to the sum of the “in LB: all ok” and “LB: need lock” counts on the “PIO Statistics--SPAM Fetches” screen.

To calculate the “total number of page requests” count, add the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics-- Data Fetches” screen to the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics--SPAM Fetches” screen.

In the previous screen examples, the buffer pool effectiveness is as follows:

$$(1,220,032 / 1,409,470) * 100 = \text{approximately } 86.5\%$$

In general, a higher percentage implies better buffering. If every page request found the page in the buffer pool, the buffer pool effectiveness would be 100%.

When a database has global buffers enabled, the global buffer versions of the “PIO Statistics--Data Fetches” and “PIO Statistics--SPAM Fetches” screens are displayed by the RMU Show Statistic utility, as shown in the following examples:

```
Node: MYNODE (1/1/2)      Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:35:33.27
Rate: 1.00 Second        PIO Statistics--Data Fetches      Elapsed: 00:04:10.48
Page: 1 of 1            DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1  Mode: Online
```

```
-----
```

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....
	max.....	cur.....	avg.....		
fetch for read	1285	220	53.6	13449	15.1
fetch for write	522	431	87.7	21986	24.8
in AS: all ok	921	611	126.0	31580	35.6
AS: lock for GB	0	0	0.0	0	0.0
AS: need lock	181	29	10.6	2662	3.0
AS: old version	0	0	0.0	0	0.0
in GB: need lock	176	11	4.2	1064	1.2
GB: old version	0	0	0.0	0	0.0
GB: transferred	0	0	0.0	0	0.0
not found: read	24	0	0.5	126	0.1
: synth	2	0	0.0	13	0.0
DAPF: success	0	0	0.0	0	0.0
DAPF: failure	0	0	0.0	0	0.0
DAPF: utilized	0	0	0.0	0	0.0
DAPF: discarded	0	0	0.0	0	0.0

```
-----
```

```
Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !
```

Node: MYNODE (1/1/2) Oracle Rdb X7.0-00 Perf. Monitor 30-JAN-1998 13:35:36.39
 Rate: 1.00 Second PIO Statistics--SPAM Fetches Elapsed: 00:04:13.60
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....
	max.....	cur.....	avg.....		
fetch for read	26	5	0.4	108	0.1
fetch for write	9	0	0.3	83	0.0
in AS: all ok	21	4	0.5	131	0.1
AS: lock for GB	0	0	0.0	0	0.0
AS: need lock	2	0	0.0	24	0.0
AS: old version	0	0	0.0	0	0.0
in GB: need lock	5	0	0.1	33	0.0
GB: old version	0	0	0.0	0	0.0
GB: transferred	0	0	0.0	0	0.0
not found: read	2	0	0.0	3	0.0
: synth	0	0	0.0	0	0.0
DAPF: success	0	0	0.0	0	0.0
DAPF: failure	0	0	0.0	0	0.0
DAPF: utilized	0	0	0.0	0	0.0
DAPF: discarded	0	0	0.0	0	0.0

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

For a database with global buffers enabled, you can calculate the following:

- The percentage of page requests that found the page in the allocate set
- The percentage of page requests that found the page in the global buffer pool
- The percentage of page requests that saved a disk I/O that would have occurred if your database had had local buffers enabled instead

You can calculate the percentage of page requests that find the page in the allocate set as follows:

$$(\text{total number found in allocate set} / \text{total number of page requests}) * 100$$

To calculate the “total number found in allocate set” count, add the sum of the “in AS: all ok”, “AS: lock for GB” and “AS: need lock” counts on the “PIO Statistics--Data Fetches” screen to the sum of the “in AS: all ok”, “AS: lock for GB” and “AS: need lock” counts on the “PIO Statistics--SPAM Fetches” screen.

To calculate the “total number of page requests” count, add the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics--Data Fetches” screen to the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics--SPAM Fetches” screen.

In the previous global buffer screen examples, the percentage of page requests that find the page in the allocate set is as follows:

$$(34,397 / 35,626) * 100 = \text{approximately } 96.5\%$$

You can calculate the percentage of page requests that find the page in the global buffer pool (this includes the pages found in the allocate set) as follows:

$$\left(\frac{\text{total number found in global buffer pool}}{\text{total number of page requests}} \right) * 100$$

To calculate the “total number found in global buffer pool” count, add the sum of the “in AS: all ok”, “AS: lock for GB”, “AS: need lock” and “In GB: need lock” counts on the “PIO Statistics--Data Fetches” screen to the sum of the “in AS: all ok”, “AS: lock for GB”, “AS: need lock” and “In GB: need lock” counts on the “PIO Statistics--SPAM Fetches” screen.

To calculate the “total number of page requests” count, add the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics--Data Fetches” screen to the sum of the “fetch for read” and “fetch for write” counts on the “PIO Statistics--SPAM Fetches” screen.

In the previous global screen examples, the percentage of page requests that found the page in the global buffer pool (including the allocate set) is as follows:

$$(35,494 / 35,626) * 100 = \text{approximately } 99.6\%$$

After calculating the allocate set effectiveness and buffer pool effectiveness, you can determine whether the number of global buffers allocated to each process (allocate sets) and the number of buffers in the buffer pool are appropriate. If you want to increase the buffer pool effectiveness for the database, and you have additional memory capacity on the system, increase the number of buffers in your buffer pool. This should increase the buffer pool effectiveness.

If the buffer pool effectiveness is satisfactory but the allocate set effectiveness is low, you can increase the number of buffers allocated to processes, which should increase the allocate set effectiveness. Performance is better if a page is found in a user's allocate set instead of in the global buffer pool. Be careful not to increase the size of allocate sets too much. For example, suppose a database has a global buffer pool of 1000 buffers, the default allocate set size for each process is 30 buffers, and 20 user processes must be able to access the database. With the 1000 buffers in the buffer pool and 20 processes that need to access the database, you should not make the default allocate set size greater than 50 buffers; when an allocate set size of 50 buffers is defined, each of the 20 user processes that needs to access the database can use all 50 buffers allocated to it.

If your database has global buffers enabled, you might wonder what percentage of your page requests saved a disk I/O that would have occurred if your database had local buffers enabled instead. To calculate the percentage of page requests that would have resulted in I/O operations if your database had had local buffers enabled instead of global buffers enabled, use the following formula. For each field in the formula,

provide the sum of the counts for the field from the “PIO Statistics--Data Fetches” and “PIO Statistics--SPAM Fetches” screens:

$$\left(1 - \frac{\text{“AS: old version”} + \text{“in GB: old version”} + \text{“not found: read”} + \text{“: synth”}}{\text{“AS: old version”} + \text{“in GB: old version”} + \text{“not found: read”} + \text{“: synth”} + \text{“in GB: need lock”}} \right) * 100$$

In the previous global buffer screen examples, the percentage of page requests that saved a disk I/O that would have occurred if the database had local buffers enabled is as follows:

$$(1 - (142 / 1239)) * 100 = \text{approximately } 99.8\%$$

To determine the number of global buffers that you will need to achieve a certain buffer pool effectiveness for a database, increase the size of the buffer pool by different amounts and calculate the percentage of page requests that find the page in the buffer pool for each increasingly larger global buffer pool.

When you set the RDMSBIND_BUFFERS logical name, you need to address several system issues. First, you may need to update the UAF quotas for the account from which the database application will be run. DIOLM must be greater than or equal to the number of database buffers. ASTLM should be at least 12 greater than DIOLM. ENQLM may need to be increased, because more locks will be used for the additional buffers. Memory parameters BYTLM and PGFLQUOTA may need to be increased to handle the increased memory requirements. Additionally, working set parameters WSDEFAULT, WSQUOTA, and WSEXTENT may need to be adjusted to allow the processes to have more memory. If increasing the number of buffers causes a large increase in the page fault rate, adjusting the working set limit for the process should resolve the problem. Finally, the SYSGEN parameter VIRTUALPAGECNT may need to be increased before the process can take advantage of the additional buffers. If increasing the number of buffers does not result in an increased consumption of virtual memory, chances are good that you have reached the VIRTUALPAGECNT limit.

If the RDMSBIND_BUFFERS logical name has already been optimized, you can use the RDMS\$BIND_WORK_VM logical name to reduce I/O activity. The RDMS\$BIND_WORK_VM logical name can be used to reduce disk I/O operations for database matching operations by allowing you to specify the amount of virtual memory in bytes that will be allocated to your process for use in matching operations. The default is 10,000 bytes. Normally, when the 10,000 bytes of virtual memory is exhausted, Oracle Rdb uses a temporary file for any additional data. Specifying larger values can reduce I/O by eliminating or reducing the need for these temporary files to be created.

The value of RDMS\$BIND_WORK_VM logical name should not be greater than the UAF parameter PGFLQUOTA.

Compare the data page request statistic to the SPAM page request statistic. If the SPAM fetch rate is high relative to the data fetch rate, it might simply be an accurate reflection of the data access patterns. However, because it might also be an indication of contention problems, you should try to determine the cause of a high number of SPAM fetches.

Look at the pool overflow statistic. If there are many pool overflows for each transaction, you might need to adjust the buffer size or the number of buffers allocated for each user. You may also want to consider enabling global buffers.

Look at the unmark buffer statistic and check the pool overflow and lock conflict statistics. If the buffer had to be unmarked due to a pool overflow or a lock conflict, Oracle Rdb is doing extra work. This may be the result of lock contention or a buffer pool being too small.

Compare the data file reads statistic to the data file writes statistic. Generally, this ratio just gives you information about the current transaction mix.

IDENTIFYING
EXCESSIVE I/O IN
STORAGE AREAS

If you detect the problem that Oracle Rdb is checking excessive pages while attempting to insert a row, you can use the “File IO Overview” screen to identify which storage area is exhibiting the behavior. The cause of the problem could be locked free space, incorrect SPAM thresholds, unique indexes, or estimated record sizes.

For high volume, transaction processing applications or applications with a large number of storage areas, it is not practical to manually examine the “IO Statistics (by file)” screen for each storage area in the database, trying to identify the particular storage area with excessive read I/O operations.

To help you more quickly identify the storage area or areas in which excessive page checks are occurring, use the “File IO Overview” screen from the “IO Statistics (by file)” submenu.

The “File IO Overview” screen shows the synchronous and asynchronous read and write I/O counts, and the database pages discarded for all storage areas, the after-image journals, the recovery-unit journal (.RUJ) file, the .ACE file, the database root file and, finally, all data and snapshot areas combined.

For example, consider the following “File IO Overview” screen:

```
Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 10-FEB-1998 12:43:13.47
Rate: 1.00 Second File IO Overview (Total Checked) Elapsed: 02:07:37.84
Page: 1 of 1 DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online
```

```
-----
```

File/Storage.Area.Name.....	Sync.Reads	SyncWrites	AsyncReads	AsyncWrites	PgDis
All data/snap files	330821	46661	0	43605	100k
data EMPIDS_OVER	270045	6331	0	31918	100k
data MF_PERS_DEFAULT	39062	1841	0	11687	741
data EMPIDS_LOW	3	3	0	0	4
ACE (AIJ Cache Electronic)	0	0	0	0	0
RUJ (Recovery-Unit Journal)	2	6	0	13230	0
AIJ (After-Image Journal)	1135	14432	0	0	0
Database Root	42	261	0	20268	0
data DEPARTMENTS	0	0	0	0	0
snap DEPARTMENTS	0	0	0	0	0
snap EMPIDS_LOW	2	3	0	0	0
data EMPIDS_MID	0	0	0	0	0
snap EMPIDS_MID	0	0	0	0	0
snap EMPIDS_OVER	13041	23439	0	0	0
data EMP_INFO	0	0	0	0	0
snap EMP_INFO	0	0	0	0	0
data JOBS	0	0	0	0	0
snap JOBS	0	0	0	0	0
snap MF_PERS_DEFAULT	8922	15044	0	0	0
data MF_PERS_SEGSTR	75	0	0	0	0
snap MF_PERS_SEGSTR	0	0	0	0	0
data SALARY_HISTORY	0	0	0	0	0
snap SALARY_HISTORY	0	0	0	0	0

```
-----
```

```
Config Exit Filter Help Menu >next_page <prev_page Options Reset Set_rate Write
```

This example shows a “File IO Overview” screen that has been configured to display the synchronous and asynchronous read and write counts, sorted by descending “total pages discarded”. Database storage areas are identified by the prefix “data” and snapshot areas are identified by the prefix “snap”. Storage areas that are added to the database are automatically shown on the screen.

Note that the EMPIDS_OVER storage area has an exceedingly large number of pages discarded, as well as an extraordinary number of synchronous read I/O operations. This storage area warrants additional investigation.

The “PgDis” column indicates the number of database pages discarded for each area, and summarizes this information for all data/snap areas.

The “File IO Overview” screen name in the display header shows the display configuration you selected.

CHECKING VERB
 FAILURES

Compare the verb successes statistic to the verb failures statistic. A high failure rate compared to the success rate indicates that many things are failing. The problem might be too many deadlocks, but is really application-dependent.

CHECKING
 SNAPSHOTS

Snapshots are another area to consider when you attempt to reduce I/O.

If the database is being used in a single-user application that does not require concurrent access to the information, then disable the snapshots. Do not incur the overhead of writing to the snapshot file if it is not necessary.

The default is that snapshot files are enabled immediate. This is probably the correct setting for snapshot files in an interactive, multi-user environment. When snapshots are enabled immediate, writers incur the overhead of updating the snapshot file every time they write to the database. When many users are accessing the data, performance will normally be optimal using immediate snapshots. This option trades additional I/O for reduced contention.

If snapshots are required and I/O resources are already limited on the disk where the storage area resides, consider moving the snapshot file to a less utilized device. If possible, place the storage area files (.RDA), snapshot files (.SNP), database rootfile (.RDB), and after-image journals so that each file type is on a separate disk.

Deferred snapshots can be used for those environments where it is acceptable for readers to be temporarily delayed. If your database is update-intensive with short transactions, and readers are not constantly using the database, you can benefit from deferred snapshots.

With deferred snapshots, read/write transactions write to the snapshot file only if a read-only transaction is in progress when the write transaction begins. The potential delay for the read-only transaction occurs when a write transaction is active when the read-only transaction starts. The read-only transaction must wait for all active write transactions to complete. New writers, who start a transaction after the read-only transaction begins, write before-images of the rows they are about to update to the snapshot files (.SNP) file.

The benefit of using snapshots enabled deferred is reduced I/O; if no read-only transactions are active, writers need not write to the snapshot file.



Be sure to review the “Snapshot Statistics” screen for additional runtime snapshot information.



Look at the retrieved record, the fetched line, and the read snap page statistics. If the fetched line number is much higher than that for retrieved record, read-only transactions are doing a lot of work to access their records. The problem might be that the access paths are not optimal. The read snap page also indicates the amount of database activity. If the read snap page number is higher than the number for retrieved record, the records fetched may be those that have been frequently updated. A

zero indicates that read-only transactions are rarely accessing data modified by update transactions.



Check the page too full statistic, which applies to read/write transactions. This statistic indicates that some data pages contain frequently-accessed data that form chains of snapshot pages.



Look at the page in use or page conflict statistics. These statistics indicate contention within the snapshot file. The problem might be that the snapshot file should be extended. The file might not extend itself because it can resolve the contention by doing extra work. However, you might want to extend the snapshot file manually to see if it helps alleviate the problem.

Analyzing Lock Resources

Locking can have a significant impact on database application response times. Locking is often the cause of intermittent performance problems where the application exhibits response time delays, yet CPU, I/O, and memory resources are still available.

There are many causes of locking problems. An application may be holding a shared resource with too restrictive a lock, or a resource may have too many users contending for it. These problems will limit the maximum database throughput that can be achieved.

If contention is evident in the database, for example, if lots of lock resources are being used up, or if many applications are stalled in the RMU Show Statistic utility “Stall Messages” screen, or if the RMU Show Statistic utility “Summary Locking Statistics” screen shows many blocking ASTs, then turning off adjustable lock granularity (ALG) should reduce the number of blocking ASTs by allowing Oracle Rdb to use the lowest level of locking immediately.

An increase in blocking ASTs may also be related to the carry-over lock optimization.

Examine the DECLARE TRANSACTION and SET TRANSACTION statements in your application to ensure that they allow maximum concurrency. If your transaction is only reading a table, set the transaction to read-only and the access mode to shared read (specify the SQL SET TRANSACTION READ ONLY RESERVING table-name FOR SHARED READ statement for the transaction). In SQL, the default is “shared read/write”. Thus, if your application uses the default, it will hold more protective locks on the resources than are necessary if the transaction is a read-only transaction. Symptoms of this problem include increased contention and lower concurrency than expected.

Node: MYNODE (1/1/1) Oracle Rdb X7.0-00 Perf. Monitor 23-JAN-1998 13:32:40.03
 Rate: 1.00 Second Summary Locking Statistics Elapsed: 00:00:04.15
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....
	max.....	cur.....	avg.....		
locks requested	0	0	0.0	0	0.0
rqsts not queued	0	0	0.0	0	0.0
rqsts stalled	0	0	0.0	0	0.0
rqst timeouts	0	0	0.0	0	0.0
rqst deadlocks	0	0	0.0	0	0.0
locks promoted	0	0	0.0	0	0.0
proms not queued	0	0	0.0	0	0.0
proms stalled	0	0	0.0	0	0.0
prom timeouts	0	0	0.0	0	0.0
prom deadlocks	0	0	0.0	0	0.0
locks demoted	0	0	0.0	0	0.0
locks released	0	0	0.0	0	0.0
blocking ASTs	0	0	0.0	0	0.0
stall time x100	0	0	0.0	0	0.0
invalid lock block	0	0	0.0	0	0.0

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Unreset Write X_plot

If the locking problems are intermittent, an application may be accessing the database sequentially instead of using an index. When an application starts a read/write transaction in shared write mode and the access is sequential, the shared write lock is promoted to protected write (PW). This has the effect of locking the whole table, and could account for intermittent locking problems. To see if an application is using the expected index, use the RDMSSDEBUG_FLAGS logical name. Setting the logical name to "S" results in the display of the retrieval strategy.

Another area that may be degrading performance is excessive deadlocks. Deadlocks can be caused by applications or by poor application design.

If your application is prone to deadlocks, you can adjust the SYSGEN parameter DEADLOCK_WAIT to detect the situation more quickly. The default value is 10 seconds. Lower the parameter if the environment is deadlock prone.

By lowering the parameter, deadlocks are detected more quickly and applications are delayed for less time. The cost of this is increased overhead to search the lock timeout queue. If you get very few deadlocks, increasing this parameter causes true deadlock situations to be undetected longer, but reduces the overhead of searching for deadlocks that exist only rarely. The more locks that are allocated, the more time it takes to search the queue and the greater the overhead to search for a deadlock condition.

Note that lowering the DEADLOCK_WAIT parameter only helps you to detect deadlocks more quickly.

If many deadlocks are encountered in a database, you can fix some of them by changing the database or applications. Some of the changes you could make to the database to fix deadlocks include reducing the size of sorted index nodes or converting a sorted index to a hashed index. Changes that you could make to applications include using update only cursors or using a restricted mode (such as protected write mode) for transactions (using a restricted mode may fix the deadlock, but could cause contention problems while the transaction is executing). In some cases, deadlocks are unavoidable.

Look at the “rqsts not queued”, “rqsts stalled”, and “rqst deadlocks” statistics. High numbers for these statistics (relative to the norm for your database) indicate contention problems. The problem might be that your database design is not optimal. The problem could also be a high number of blocking asynchronous system traps (ASTs) or it may be that users are contending for frequently-accessed data.

Analyzing Journal Resources

CHECKING AIJ
 JOURNALS

If after-image journaling is enabled, and the disk to which the after-image journal is being written is highly utilized, one tuning possibility is to move the after-image journal to a less utilized disk. You cannot change the after-image journal location using logical names or configuration parameters. Instead, use the SQL ALTER DATABASE statement to move the after-image journal. Use the JOURNAL ALLOCATION IS and JOURNAL EXTENT IS clauses to further enhance performance of AIJ writes. These clauses ensure that the initial allocation of the journal file is large enough so that transactions do not need to wait for extents to occur.

Another reason to move the after-image journal to its own disk or to a non-database disk is that doing this makes your database more reliable. The after-image journal allows the DBA to roll forward the database from the point of the last backup when a disk failure has occurred on one of the disks where an storage area files (.RDA), database rootfile (.RDB), or snapshot files (.SNP) resided. For this reason, the after-image journal should be on a separate disk from the storage area files, database rootfile, and snapshot files for the database.

Ideally, the after-image journal should be on a controller separate from all other database files.

If you use multiple after-image journals for your database, each after-image journal should be on a separate disk from the database's storage area files (.RDA), database rootfile (.RDB), and snapshot files (.SNP). Otherwise, if a disk failure occurs, you may not be able to recover.

Oracle Rdb recommends that after-image journaling always be enabled unless the database is read-only or it does not need to be recovered in case of failure.

You can improve database performance in some environments by enabling fast commit processing. When you enable the Fast Commit feature, Oracle Rdb keeps updated pages in the buffer pool and does not write the pages to disk when a transaction commits. Updated pages remain in the buffer pool until they are written to the database as part of an asynchronous batch-write operation. Also, when a user-specified threshold (called a checkpoint) is reached, all the updated pages for multiple transactions are flushed to disk. If a transaction fails, all the previous transactions back to the last checkpoint must be redone because their updated pages have not been written to disk. Oracle Rdb uses information written to the after-image journal to redo these transactions.

You should consider enabling fast commit transaction processing when you have a stable transaction processing environment and the following conditions exist: a process updates the same rows for multiple transactions; or, transactions are short and do not update many pages. Typically, these types of transactions tend to have few buffer pool overflows. The I/Os to the database are reduced for these types of transactions, which means the redo operation performed when a transaction fails and fast commit transaction processing is enabled will be shorter than the redo operation for long transactions that update many pages.

When you enable fast commit transaction processing, you can also enable the Commit To Journal Optimization feature. This option can provide a significant increase in commit processing speed by eliminating the majority of I/O to the database root. This option enhances performance in database environments that are update-intensive.

Compare the records written statistic to the blocks written statistic. The ratio of records written to blocks written gives an indication of the size of the AIJ records (or the size of the record modified).

CHECKING RUJ
 JOURNALS

Compare the recovery-unit journal (.RUJ) file read and write statistics. Too many .RUJ file reads indicate too many rollbacks. Too many .RUJ file writes might indicate contention problems or transactions that modify a lot of data. More than one write for each transaction indicates that marked pages are being written back to the database before the actual commit. This means either another recovery-unit requested the page, the buffer pool overflowed, or there are very large transactions with lots of updates.

AIJ STATISTICS
 SCREEN

The “AIJ Statistics” screen monitors both logical and physical after-image journaling activity.

The following is an example of an "AIJ Statistics" screen:

Node: MYNODE (1/1/16) Oracle Rdb X7.0-00 Perf. Monitor 29-JAN-1998 11:02:24.65
 Rate: 1.00 Second AIJ Statistics Elapsed: 00:26:45.61
 Page: 1 of 1 DISK\$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1 Mode: Online

statistic..... name.....	rate.per.second.....			total..... count.....	ave rage..... per.trans....
	max.....	cur.....	avg.....		
AIJ file writes	16	1	8.4	13553	1.0
data	14	2	7.3	11840	0.8
control	0	0	0.0	0	0.0
file extend	0	0	0.0	0	0.0
switch over	0	0	0.0	0	0.0
tx. block span	171	42	84.8	136241	10.1
records written	84	16	47.0	75490	5.6
blocks written	32	3	17.8	28726	2.1
filler bytes	6616	948	2266.1	3638622	272.1
lock rebuilds	0	0	0.0	1	0.0
AIJ file reads	19	0	0.4	694	0.0
shuffle averted	0	0	0.0	0	0.0
suspended switch	0	0	0.0	0	0.0

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

AIJ JOURNAL
 INFORMATION
 SCREEN

The "AIJ Journal Information" screen provides online information about all of a database's after-image journals on the current node. Most of the information displayed on the screen is obtained in real time, which means that the screen is automatically updated as AIJ database parameters are modified, or as AIJ operations such as a backup or journal switch-over are performed.

The following is an example of the “AIJ Journal Information” screen:

```

Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:37:46.90
Rate: 3.00 Seconds      AIJ Journal Information           Elapsed: 03:38:13.55
Page: 1 of 3           SQL_DISK1:[RICK.V70]MF_PERSONNEL.RDB;1       Mode: Online
-----
Journaling: Enabled   Shutdown: 60   Notify: Disabled   State: A ccessible
ALS: Manual          ABS: Disabled ACE: Disabled   FC: Enabled   CTJ: Enabled
ARB.Count: 300 ARB.Avail: 298
After-Image.Journal.Name..... SeqNum AIJsize CurrEOF Status. State.....
RICK1                Unused    512   Empty Latent  A ccessible
RICK2                Unused    512   Empty Latent  A ccessible
RICK3                Unused    512   Empty Latent  A ccessible
RICK4                19       513   2 Current A ccessible
RICK5                Unused    512   Empty Latent  A ccessible
RICK6                18 *BACKUP NEEDED* Written A ccessible
Available AIJ slot 1
Available AIJ slot 2
Available AIJ slot 3
Available AIJ slot 4
Available AIJ slot 5
Available AIJ slot 6
Available AIJ slot 7
Available AIJ slot 8
-----
Bell Exit Help Menu >next_page <prev_page Refresh Set_rate Write Zoom !
  
```

The “AIJ Journal Information” screen contains information relating to after-image journaling in general and information on each individual journal, including reserved after-image journal slots.



Note that information on the “AIJ Journal Information” screen is for the current node only. Because an after-image journal is accessed by all nodes modifying the database, the information for one node could become stale. Therefore, the “Refresh” on-screen menu option is provided. The “Refresh” option causes the current node's information to be synchronized with all other nodes accessing the database.

CHECKPOINT
 STATISTICS
 SCREEN

The “Checkpoint Statistics” screen shows transaction and checkpoint activity. Statistics are displayed for all processes on the node for a particular database. The total number of checkpoints, with a breakdown of the reasons for checkpointing, is displayed. The sum of all checkpoint intervals is also displayed, using three different metrics. You can use this information to compute the average interval between checkpoints, allowing you to decide if a checkpointing interval should be adjusted, and by how much.

Some of the columns provided in the “Checkpoint Statistics” screen measure events on a per second or per transaction basis. These columns are useful for measuring frequently-occurring events such as I/O operations. Because checkpointing typically occurs at a slower rate, you will find the most meaningful checkpointing information in the total count column of the “Checkpoint Statistics” screen. Oracle Corporation

recommends that you refer to this column when you use checkpoint statistics to determine optimal checkpoint intervals.

The following is an example of a "Checkpoint Statistics" screen:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 29-JAN-1998 14:51:22.60
Rate: 1.00 Second      Checkpoint Statistics           Elapsed: 04:15:43.56
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
```

```
-----
statistic.....      rate.per.second..... total..... ave rage.....
name.....           max..... cur..... avg..... count..... per.trans....
transactions         3         0         2.6        40969        1.0
checkpoints          0         0         0.0         655         0.0
  AIJ growth         0         0         0.0         0         0.0
  txn limit          0         0         0.0         0         0.0
  time limit         0         0         0.0         0         0.0
  rollback           0         0         0.0         0         0.0
  AIJ backup         0         0         0.0        179         0.0
  global             0         0         0.0        472         0.0
interval: AIJ blks   16         0        12.8     197254        4.8
interval: tx count   3         0         2.6     40668         0.9
interval: seconds    4         0         3.2     50379         1.2

checkpoint stall     1         0         0.8     13215         0.3
flushed buffers      0         0         0.1      3029         0.0
-----
```

Exit Graph Help Menu Options Pause Reset Set_rate Time_plot Write X_plot Yank !

Keep in mind that checkpointing influences recovery time. The main reason to consult checkpoint statistics is to find the average interval per checkpoint. You can use the information in the total count column to compute this average. For each category of checkpoint reason, use the average interval per checkpoint to help you decide if a checkpointing interval should be adjusted, and by how much.

You can set checkpoint limits for the following:

Limit	Description
AIJ file growth	You specify the .AIJ file growth limit by using the CHECKPOINT INTERVAL IS n BLOCKS clause of the ALTER DATABASE statement. If CHECKPOINT INTERVAL IS 1000 BLOCKS is specified, each process checkpoints when the .AIJ file has grown at least 1000 blocks since the process' last checkpoint.
Time	You specify the time limit by using the CHECKPOINT TIMED EVERY n SECONDS clause of the ALTER DATABASE statement. If CHECKPOINT TIMED EVERY 600 SECONDS is specified, each process checkpoints when at least 10 minutes have elapsed since the process' last checkpoint.

<u>Limit</u>	<u>Description</u>
Transactions	You specify the transaction limit by using the RDMSBIND_CKPT_TRANS_INTERVAL logical name. If RDMSBIND_CKPT_TRANS_INTERVAL is defined as a system logical name set to the value 10, each process checkpoints after 10 transactions unless a user redefines the RDMSBIND_CKPT_TRANS_INTERVAL logical name to a different value. That is, if a user defines RDMSBIND_CKPT_TRANS_INTERVAL as a process logical name and sets a value of 5, that user checkpoints after 5 transactions.

To find the average interval for a checkpoint, divide each of the interval categories by the total number of checkpoints. In the example, the seconds interval count is 50,379 and the total number of checkpoints is 655, so the average number of seconds between each time-triggered checkpoint is approximately 77.

Likewise, the transactions interval is 40,668 and the total number of checkpoints is 655, so the average number of transactions between checkpoints is approximately 63.

However, when computing the average interval in AIJ blocks, you must divide the AIJ block interval by the total number of checkpoints minus the number caused by AIJ backups. Although checkpoints caused by AIJ backups are counted in the total number of checkpoints, they are not counted in the total of AIJ block intervals. The following calculation shows that the .AIJ file grows by an average of approximately 415 blocks between each checkpoint:

$$\text{Avg AIJ Block Interval} = 197,254 / (655 - 179) = 414.399$$



Note that there may be more reasons for checkpoints than there are total checkpoints. This can occur because a single checkpoint may be triggered by more than one event. For example, a checkpoint may occur because of time and after-image journal growth. Although both columns will increment by one, the total checkpoint column only increments by one.

Analyzing Memory Resources

EXCESSIVE PAGE
 FAULTS³²

Excessive hard or soft page faulting can be a limiting factor of process performance. The “Process Accounting” screen is the best RMU Show Statistic utility screen available to determine if an application process is experiencing excessive page faults.

³² Contributed by Norm Lastovica.

Sometimes page faulting occurs during Oracle Rdb sort operations. Common causes of sorts include the SQL GROUP BY, ORDER BY, UNION and DISTINCT clauses specified for a query, and index creation operations. Defining the logical name RDMSS\$DEBUG_FLAGS to "RS" can help determine when Oracle Rdb sort operations are occurring and to display the sort keys and statistics.

Oracle Rdb includes its own copy of the OpenVMS SORT32 code within the Oracle Rdb images and does not generally call the routines in the OpenVMS runtime library. Oracle Rdb contains a copy of the SORT32 code to provide stability between various versions of Oracle Rdb and OpenVMS. Also, Oracle Rdb calls the sort routines from executive processor mode, which is difficult to do using the SORT32 sharable image.

Other database utilities, such as IMPORT and RMU Load, call the OpenVMS SORT runtime library directly.

At the beginning of a sort operation, the SORT code allocates virtual memory for working storage. The sort operation uses this memory for temporary buffers, in-memory copies of the data, and sorting trees. The sort operation does not directly consider the process' quotas or parameters when allocating this memory.

The effects of WSQUOTA and WSEXTENT are indirect. At the beginning of each sort operation, the sort operation attempts to adjust the process working set to the maximum possible size. The sort operation requests a working set limit of %X7FFFFFFF pages, which is the maximum possible. The sort operation then uses a value of 75% of the returned actual working set for virtual memory scratch space. The allocated scratch space is then initialized and the sort begins.

It is the initialization of the scratch space that generally causes page faults while accessing the pages newly added to the working set. Pages that were in the working set already may have been faulted out as the new pages were faulted in. Once the sort operation completes, the pages that had been faulted out of the working set are likely to be faulted back into the working set.

When a process' working set is limited by the working set quota (WSQUOTA) parameter and the working set extent (WSEXTENT) parameter is a much larger value, the first call to the sort routines can cause many page faults as the working set grows. Using a value of WSEXTENT that is closer to WSQUOTA can help reduce the impact of this case.

With some OpenVMS versions, the AUTOGEN utility sets the SYSGEN parameter PQL_MWSEXTENT equal to the WSMAX parameter. This means that all processes on the system end up with WSEXTENT the same as WSMAX. Since that might be quite high, sorting might result in excessive page faulting. You may want to explicitly set PQL_MWSEXTENT to a lower value if this is the case on your system.

Sort work files are another factor to consider when tuning for Oracle Rdb sort operations. When the operation can not be done in the available memory, the sort operation will use temporary disk files to hold the data as it is being sorted. The *Oracle Rdb*

Guide to Performance and Tuning contains more detailed information about sort work files.

The logical name `RDMS$BIND_SORT_WORKFILES` specifies how many work files the SORT utility is to use if work files are required. The default is 2 and the maximum number is 10. The work files can be individually controlled using the `SORTWORK n` logical names, where n is the value “0” through “9”. You can increase the efficiency of sort operations by assigning the location of the temporary sort work files to different disk devices. These assignments are made by using up to ten logical names, `SORTWORK0` through `SORTWORK9`.

Normally, the sort operation places work files in the user's `SYSSCRATCH` directory. By default, `SYSSCRATCH` is the same device and directory as the process “home” location. Spreading the I/O load over many disks improves efficiency as well as performance by taking advantage of the system resources and helps prevent disk I/O bottlenecks. Specifying that a user's work files will reside on separate disks permits overlap of the SORT read/write cycle. You may also encounter cases where insufficient space exists on the `SYSSCRATCH` disk device; for example, while Oracle Rdb builds indexes for a very large table. Using the `SORTWORK0` through `SORTWORK9` logical names can help you avoid this problem.

Note that the sort operation uses the work files for different sorted runs, and then merges the sorted runs into larger groups. If the source data is mostly sorted, then not every sort work file may need to be accessed. This is a possible source of confusion because even with 10 sort work files, it is possible to exceed the capacity of the first sort work file and the sort operation will fail never having accessed the remaining 9 sort work files.

Note that the logical names `RDMS$BIND_WORK_VM` and `RDMS$BIND_WORK_FILE` do not affect or control the sort operation. These logical names are used to control other temporary space allocation within Oracle Rdb.

Reviewing the “Process Accounting” screen is the best means to observe excessive page faulting. The following “Process Accounting” screen example shows a process with possibly excessive page faulting:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor 27-JAN-1998 10:37:46.90
Rate: 3.00 Seconds      Process Accounting              Elapsed: 00:20:11.30
Page: 1 of 1           SQL_DISK1:[RICK.V70]MF_PERSONNEL.RDB;1      Mode: Online
-----
```

```
Process.ID Process.name... CPUtime... EnqCnt. PGflts. NumDio. WSsize. VMsize.
6E2012D3:1 RICK          00:00:15.46  17809  90775   808   4471  15082
6E201887:1 SMITH         00:00:11.23  17882  17383   444   3606  14426
```

```
-----
Exit Full Help Menu >next_page <prev_page Set_rate Write Zoom !
```

ANALYZING
 VIRTUAL MEMORY
 OPTIONS³³

Oracle Rdb on OpenVMS includes several options designed to help improve performance and allow better management of a system's physical and virtual memory. These options include:

- Locking local buffers into physical memory
- Placing shared buffers into system virtual address space
- Very Large Memory (VLM)

This section describes how these options interact with OpenVMS virtual and physical memory usage and outlines some tuning information if you are looking to implement these capabilities.

OpenVMS Virtual Memory

A virtual address is a 32-bit or 64-bit integer identifying a byte location in virtual address space. The memory management hardware translates a virtual address into a physical address. Current versions of OpenVMS are 32-bit operating systems; that is, they use 32 bit virtual addresses. Furthermore, OpenVMS Alpha V7.0 added the ability to access and manipulate a 64-bit address space.

The OpenVMS virtual memory system provides each process with a 4-billion byte (4GB) address space. This virtual address space is divided into two parts: the system address space and the process address space. The system address space is the same for all processes. It contains the operating system code and data structures. Each process has its own separate process address space. However, processes can share memory; the physical memory exists in multiple virtual address spaces.

³³ Contributed by Norm Lastovica.

The upper half of virtual address space (2GB with addresses 80000000 through FFFFFFFF) is termed system space. All processes use the same address translation map for system space, so system space is shared among all processes. System space is sometimes referred to as S0 and S1 space.

The lower half of the virtual address space (2GB with addresses 00000000 through 7FFFFFFF) is termed per-process space. Each process has a separate translation map for per-process space, so the per-process spaces are unique and unshared.

The per-process space is further divided in half with the lower half of the per-process space (1GB with addresses 00000000 through 3FFFFFFF) called the program area or P0 space and with the upper half (1GB with addresses 40000000 through 7FFFFFFF) called the control area or P1 space. Usually, programs and program data is stored in P0 space. P1 space contains the process's stacks (1 each for kernel, executive, supervisor and user modes), along with DCL and RMS data.

OpenVMS Alpha version 7 adds support for 64-bit virtual addressing, which makes the 64-bit virtual address space defined by the Alpha architecture available to the OpenVMS Alpha operating system and to application programs. The 64-bit addressing features allow processes to map and access data beyond the limits of 32-bit virtual addresses. Both process-private and system virtual address space now extend well beyond 2GB. The new portions of process and system virtual memory are called P2 space and S2 space respectively and they are huge.

Locking Local Buffers When using local buffers on OpenVMS Alpha systems, you can chose to lock them into physical memory using OpenVMS buffer objects. Locking Oracle Rdb local buffers into memory can increase symmetric multiprocessing (SMP) parallelism and scaling and improves I/O performance by eliminating OpenVMS overhead.

To utilize this option you must: do the following items:

- Run Oracle Rdb along with OpenVMS Alpha Version 6.1 or higher.
- Define the logical name RDMSBIND_BUFOBJ_ENABLED to be the value "1".
- Increase the OpenVMS user quota BYTLM by the number of bytes in your local buffer pool.
- Ensure that the system parameter MAXBOBMEM (on OpenVMS V7.0 and later) is large enough.

To calculate the size in bytes of the buffer pool:

$$\text{size} = \text{number_of_buffers} * \text{buffer_size} * 512$$

For example, if the number of local buffers is 1000, and the buffer size is 8 blocks, determine the value of the BYTLM parameter using the following formula: 1000 * (8

* 512) = 4,096,000 bytes. Do not use this feature on systems that are memory constrained. The physical memory used by the buffer objects becomes unavailable for use by any other process until the image is terminated. The pages that constitute the buffer object (the process' local buffers) are permanently locked into physical memory (but not the process' working set) and double mapped into system space. Because the buffers are also mapped in system space, there must be enough system virtual address space to satisfy the request.

Oracle Rdb utilization of OpenVMS buffer objects is currently not available on databases with global buffers enabled.

red Memory into System Space

Another Oracle Rdb memory management option allows a database shared memory section to be placed in OpenVMS system space (the S0/S1 region). This is enabled with the "SHARED MEMORY IS SYSTEM" qualifier of the "ALTER DATABASE" statement.

By default, the database shared memory section is created and accessed in process space (the P0 region) using OpenVMS global sections. When you use global sections created in the process space, you and other users share physical memory and the OpenVMS operating system maps the section to the private address space for each user. The memory allocated to the global section is pageable and is not fixed into physical memory.

Enabling SHARED MEMORY IS SYSTEM places the shared memory section in OpenVMS Alpha system virtual address space. This section is fully resident, or pinned, in memory and does not affect the quotas of the working set of a process. On OpenVMS, system space (the S0/S1 region) is limited to 2GB of virtual address space. The Oracle Rdb system space buffers reside in this virtual address region along with much of the operating system.

System space is critical to the entire system. System space buffers are not paged; therefore, they use physical memory, reducing the amount of physical memory available for other system tasks. This may be an issue if your system is constrained by memory. You should be careful when you allocate system space. Non-paged dynamic pool (NPAGEDYN) and the cluster cache (VCC/VIOC) are some examples of system resources that also use system space. To find out if your system has enough physical and virtual memory in order to put a database's shared memory in system space, first determine the size of the database section by using the "Buffer Information Screen", located in the "Database Parameter Info" submenu..

In the following example, 12,052,308 bytes are needed to map the global section:

```
Node: MYNODE (1/1/16)   Oracle Rdb X7.0-00 Perf. Monitor   3-FEB-1998 05:39:34.73
Rate: 1.00 Second      Buffer Information              Elapsed: 00:00:16.33
Page: 1 of 1          DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1   Mode: Online
-----
```

```
Default user buffer count is 20
Default recovery buffer count is 20
Buffer size is 6 blocks
Maximum pages per buffer is 3
Global Buffers are disabled
- Global buffer count is 250
- Maximum global buffer count per user is 5
- Page transfer via memory is disabled
Global section size with global buffers disabled is 358910 bytes
- With global buffers enabled is 12052308 bytes
Asynchronous pre-fetch is enabled
- Maximum pre-fetch depth is 5 buffers
Detected asynchronous pre-fetch is enabled
- Maximum pre-fetch depth is 4 buffers
- Pre-fetch threshold is 4 pages
Asynchronous batch-write is enabled
- Clean buffer count is 5
- Maximum batch size is 4 buffers
```

```
-----
Exit Help Menu Options Refresh Set_rate Write !
```

On an OpenVMS Alpha system (where pages are 8192 bytes long), about 1472 pages would be needed for the global section. Use the OpenVMS “SHOW MEMORY/PHYSICAL” command to determine the number of physical pages of memory available (free) on the system:

```
$ SHOW MEMORY /PHYSICAL
Physical Memory Usage (pages):      Total    Free   In Use  Modified
Main Memory (2048.00Mb)             262144  204157  47427   10560
```

On this system, there is 2GB of physical memory and 204,157 pages that are free (not in use).

Further, since the shared memory will reside in system space (the S0/S1 region), there needs to be enough virtual address space remaining as well as enough contiguous system page table entries (SPT) to map the memory.

One way to help determine if there is enough virtual address space remaining to map the global section is to use the SDA command "CLUE MEMORY /LAYOUT" as in the following example:

```
$ ANALYZE/SYSTEM
SDA> CLUE MEMORY /LAYOUT
System Virtual Address Space Layout:
-----
Item                               Base           End            Length
Execlet Code Region                80000000      80800000      00800000
Resident Image Code Region         80800000      80C00000      00400000
Execlet Data Region                8D300000      8D450000      00150000
System Header                      80C00000      80C2A000      0002A000
PFN Database                       80C2A000      8142A000      00800000
Error Log Allocation Buffers       8142A000      8142C000      00002000
Nonpaged Pool (initial size)       81436000      83CF0000      028BA000
Nonpaged Pool Expansion Area       83CF0000      8D454000      09764000
Balance Slots                      8D454000      D1CCA6C8      448766C8
.
.
.
Executive Mode Data Page           D34A2000      D34A4000      00002000
System Space Expansion Region       D7800000      FFE00000      28600000
System Page Table (SPT)            FFE00000      FFFFFFFF      00200000
```

In this example there are 28,600,000 hexadecimal bytes (677,380,096 in decimal) of available system address space (this is about 82,688 pages). There still might be less physical memory available on the system than the amount of available virtual address space.

Note that on OpenVMS Alpha versions prior to V7.0, a significant amount of the system virtual address space may be consumed by balance slots. Starting in OpenVMS Alpha V7.0, the process page table entries are not located in the balance slots. This means that there will be less system space used for the balance slots. For systems running older versions of OpenVMS, reducing the number of balance slots, or reducing the system parameters VIRTUALPAGECNT and WSMAX can help reduce the amount of system space virtual memory consumed by the balance slots.

Because Oracle Rdb system space buffers do not count against a process' virtual page count, it is often a good idea to reduce the VIRTUALPAGECNT system parameter to reduce the system virtual address space used by the balance slots when running on OpenVMS Alpha versions prior to V7.0. The OpenVMS AUTOGEN procedure will generate the maximum pagelets (2GB) value for VIRTUALPAGECNT for any system with at least 2GB of memory. This results in a page table size of 2MB for each process. Multiply that by about 500 processes in the balance set and a full gigabyte of system virtual address space is consumed. Most applications will not use more than a few hundred megabytes for process space. Therefore, reducing the VIRTUALPAGECNT value potentially frees more system virtual address space for Rdb buffers.

The address space reserved for non-paged pool is based on the NPAGEVIR system parameter which is usually 4 times NPAGEDYN. If your system is already well tuned and you don't see non-paged pool expansion, a value of perhaps 1.5 * NPAGEDYN may be more appropriate. This may further increase the amount of available system space virtual addresses.

The Oracle Rdb monitor attempts to create the database global section in system space when a database is being opened. If there is not enough physical memory on the system to map the global section, a "SYSTEM-F-INSFRPGS" message will be included in the monitor log file.

For example:

- 11-NOV-1997 15:27:46.97 received user attach request from 2B6:4
- process name TTA0, user ORACLERDB
 - image name "\$1\$DKB200:[SYS0.SYSCOMMON.][SYSEXE]SQL\$70.EXE;1"
 - for database "_DKA0:[DB]MYDB.RDB;1" [_DKA0] (138,241,65)
 - number of global buffers is 33000, maximum buffers per user is 3000
 - unable to create SSB region
 - "%SYSTEM-F-INSFRPGS, insufficient Free Pages available"
 - database shutdown of "_DKA0:[DB]MYDB.RDB;1" is complete
 - sending user attach reply to requestor
 - %RDMS-F-CANTCREGBL, error creating and mapping database global section

If there is not enough system virtual address space available and there is enough physical memory, the monitor will report a "SYSTEM-F-INSFSPTS" message, as follows:

- 12-NOV-1997 13:54:46.34 received open database request from CEF:0
- process name RTA1, user ORACLERDB
 - for database "DKA0:[DB]MYDB.RDB;1" [_DKA0] (9132,143,0) number of global buffers is 33000, maximum buffers per user is 3000
 - unable to create SSB region "%SYSTEM-F-INSFSPTS, insufficient SPTEs available"
 - database shutdown of "DKA0:[DB]MYDB.RDB;1" is complete
 - sending open database reply to requestor
 - %RDMS-F-CANTOPENDB, database could not be opened as requested

On this system, there is not enough system virtual address space to create the database shared memory section in system space.

The OpenVMS documentation suggests increasing the SYSGEN parameter SPTREQ when the SYSTEM-F-INSFSPTS error occurs. However, OpenVMS Alpha does not use this parameter and it has no effect.

VLM and Row Caches Very large memory (VLM) support allows Oracle Rdb to use as much physical memory as is available on your system and to dynamically map it to the virtual address space of database users. VLM provides access to a large amount of physical memory (not limited to the 32-bit addressable 4GB) through small virtual address windows.

Even though VLM buffers are created in physical memory, the virtual address windows are defined and maintained in each user's private virtual address space.

VLM is available on OpenVMS Alpha only for the Oracle Rdb row caching feature. Consult the Oracle Rdb *Guide to Database Performance and Tuning* manual for more information about row caching.

When you create a row cache or modify a row cache definition, you have the option of specifying where, in memory, you want Oracle Rdb to create the cache. Row caches can reside in a process global section, a system space buffer or in very large memory (VLM).

When you use global sections created in the process space, you and other users share physical memory and the OpenVMS operating system maps a cache to the private address space for each user. Use the SHARED MEMORY IS PROCESS parameter to specify that the cache be created in a process global section.

The system space global section is located in the OpenVMS Alpha system space, which means that a system space global section is fully resident, or pinned, in memory and does not affect the quotas of the working set of a process. System space memory is critical to the overall system. System space buffers are not paged; they use physical memory, reducing the amount of physical memory available for other system tasks. This may be an issue if your system is constrained by memory. You should be careful when you allocate system space. Nonpaged dynamic pool (NPAGEDYN), balance slots and the Cluster Cache (VCC/VIOC) are some examples of system resources that use system space.

The Very Large Memory (VLM) feature allows Oracle Rdb to use as much physical memory as is available on your system and to dynamically map it to the virtual address space of database users. VLM provides access to a large amount of physical memory through small virtual address windows. Even though VLM is defined in physical memory, the virtual address windows are defined and maintained in each user's private virtual address space.

You view the physical memory through windows. A window is a range of virtual addresses that can be mapped to a portion of the 64-bit VLM physical memory. A window can be moved (re-mapped) to any location within the VLM physical memory. You can specify the number of window panes with the WINDOW COUNT parameter.

Oracle Rdb determines the size of each window. The minimum window size is the system page size (currently, all Alpha systems have a page size of 8192 bytes). Windows will be sized large enough to map at least one entire row in the cache.

By default, Oracle Rdb allocates 100 window panes to a process. Additional windows can help increase performance on very active systems. Windows consume virtual address space but do not, by themselves, consume much additional physical memory.

Once a row cache has been defined, the "Row Cache Information" screen can be used to verify information for a cache:

```
Node: MYNODE (1/1/1)      Oracle Rdb X7.0-00 Perf. Monitor  4-FEB-1998 06:10:12.85
Rate: 1.00 Second        Row Cache Information      Elapsed: 00:00:17.02
Page:  4 of 24  DISK$:[DB_HOMEDIR.WORK.STATS]MF_PERSONNEL.RDB;1  Mode: Online
```

```
-----
Row cache "EMPIDS_OVER"
Cache ID number is 4
Cache filespec is "_$111$DUA347:[DB_HOMEDIR.WORK.STATS]"
Row slot allocation is 100000 for maximum record length 1000
Maximum working set count is 10
Row replacement is enabled
Cache will be mapped in process space
Dynamic mapping is disabled
Dynamic mapping window count is 100
No checkpoint for sequence 0
```

```
-----
Exit Help Menu >next_page <prev_page Options Refresh Set_rate Write !
```

The RMU Dump Header utility can be used to determine the sizes of the various virtual and physical memory regions for a cache:

```
Row cache "MYCACHE"
Cache ID number is 1
Allocation...
- Row slot count is 100000
- Maximum row size allowed in cache is 1000 bytes
- Working set count is 10
- Maximum slot reservation count is 20
.
.
.
Cache-size in different sections of memory...
- Total normal or system memory requirement is 105338736 bytes
- Total very large memory requirements...
- Physical memory requirement is 105338736 bytes
- Virtual memory address space requirement is approximately 1643016 bytes
```

ARED MEMORY IS PROCESS

When the database uses the attribute SHARED MEMORY IS PROCESS, the row cache control structures are located in a process global section. The storage of the data rows depends on whether large memory is enabled or disabled. If large memory is enabled, data is stored in physical memory and windows from each user's process virtual address space are needed to access the data. If large memory is disabled, data is stored in a process global section and no windows are needed to access the data.

When the database uses the attribute `LARGE MEMORY IS DISABLED`, the storage of the data rows and the row cache control structures depends on whether shared memory is `PROCESS` or `SYSTEM`. If shared memory is `PROCESS`, the data and row cache control structures are stored in a process global section and no windows are needed to access the data. If shared memory is `SYSTEM`, the data and row cache control structures are stored in system space and no windows are needed to access the data.

When the database uses the attribute `LARGE MEMORY IS ENABLED`, the data rows are stored in physical memory and process private virtual address windows are needed to access the data. The storage of the row cache control structures depends on whether shared memory is `PROCESS` or `SYSTEM`. If shared memory is `PROCESS`, the control structures are stored in a process global section. If shared memory is `SYSTEM`, the control structures are stored in system space.

SHARED MEMORY IS SYSTEM

When the database uses the attribute `SHARED MEMORY IS SYSTEM`, the row cache control structures are stored in system space. The storage of the data rows depends on whether large memory is enabled or disabled. If large memory is enabled, data is stored in physical memory and windows from each user's process virtual address space are needed to access the data. If large memory is disabled, data is stored in system space and no windows are needed to access the data.

When the database uses the attribute `LARGE MEMORY IS DISABLED`, the storage of the data rows and the row cache control structures depends on whether shared memory is process or system. If shared memory is process, the data and row cache control structures are stored in a process global section and no windows are needed to access the data. If shared memory is system, the data and row cache control structures are stored in system space and no windows are needed to access the data.

When the database uses the attribute `LARGE MEMORY IS ENABLED`, the data rows are stored in physical memory and process private virtual address windows are needed to access the data. The storage of the row cache control structures depends on whether shared memory is process or system. If shared memory is process, the control structures are stored in a process global section.

Index

/ PROMPT_TIMEOUT, 28, 50, 144
/(NO)BROADCAST, 20
/(NO)CLUSTER, 20
/(NO)CYCLE, 21
/(NO)DBKEY_LOG, 21
/(NO)HISTOGRAM, 22
/(NO)INTERACTIVE, 22
/(NO)LOCK_TIMEOUT_LOG, 23
/(NO)LOG, 23
/(NO)LOGICAL_AREA, 23
/(NO)NOTIFY, 20, 24
/(NO)OUTPUT, 27
/(NO)PROMPT_TIMEOUT, 28
/(NO)STALL_LOG, 29
/ALARM, 19, 20, 24, 25, 117, 130
/BROADCAST, 20, 25, 130, 149
/CLUSTER, 20, 24, 32, 130, 132, 133, 136, 156, 160
/CONFIGURE, 20, 121
/CYCLE, 21, 125, 126, 127, 130
/DBKEY_LOG, 21, 30, 122
/DEADLOCK_LOG, 21, 22, 30
/HISTOGRAM, 22
/INPUT, 14, 15, 19, 20, 22, 27, 31, 32, 33, 36, 41, 42, 45, 46, 47, 110, 117, 136, 142, 145, 169, 223
/INTERACTIVE, 14, 22, 23
/LOCK_TIMEOUT_LOG, 21, 23, 30
/LOG, 23, 128
/NOBROADCAST, 56
/NOCLUSTER, 20, 132
/NOCYCLE, 21
/NODBKEY_LOG, 21
/NOHISTOGRAM, 22
/NOINTERACTIVE, 15, 28, 29, 31, 32
/NOLOCK_TIMEOUT_LOG, 23
/NOLOG, 23
/NOLOGICAL_AREA, 102
/NONOTIFY, 24
/NOOUTPUT, 27
/NOSTALL_LOG, 29
/NOTIFY, 20, 24, 25, 130, 156, 157
/OPTION, 29, 44, 162

/OPTIONS, 25, 26, 27
/OUTPUT, 14, 15, 22, 23, 25, 27, 28, 30, 32, 36, 45, 46, 47, 110, 136, 142, 169, 222, 223
/PROMPT_TIMEOUT, 28, 50
/REOPEN_INTERVAL, 28, 35
/RESET, 28
/SCREEN, 28, 29, 109
/STALL_LOG, 26, 29, 30, 122, 214
/STATISTICS, 12, 13, 36, 38
/TIME, 14, 15, 29, 30, 32, 33
/UNTIL, 8, 14, 30, 31, 32, 160
"Lock stall threshold, 205
ACTIVE, 75
Adjustable Locking Granularity, 205
AIJ Backup Server, 75, 144, 165, 182, 183
AIJ cache, 186, 187
AIJ Log Server, 144, 146, 150, 165, 180, 181, 185, 186
AIJ Request Block, 184
AIJ Request Blocks, 181, 184, 186
AIJ_ARBS_PER_IO, 185
AIJ_BKGRD_ARB_RATIO, 186
AIJ_BLKs_PER_IO, 185
AIJ_CACHE_OVF_RATIO, 187
AIJ_NETWORK_BPS, 189
AIJ_SEC_TO_EXTEND, 182
AIJDB, 76
AIJFB, 76
ARBs per AIJ I/O, 185
Asynchronous Batch-Write, 172
Asynchronous Pre-Fetch, 171
binary file replay, 22
Blocks per AIJ I/O, 185, 186
BTR_FETCH_DUP_RATIO, 207
BTR_LEF_FETCH_RATIO, 207
BTR_STORE_DUP_RATIO, 207
B-tree duplicate fetch threshold, 207
B-tree duplicate store threshold, 207
B-tree leaf node store threshold, 207
Cache overflow threshold, 187
CLIENT, 77
Cluster Statistics Collection and Presentation, 131, 132, 134
Commit to Journal Optimization, 229, 245
Configuration File, 151
CPT, 77
Dashboard Console, 161, 162
Database Dashboard, 164, 166, 169
Database Recovery Process, 150, 164, 173, 187, 192, 193
DBApack, 6
DBKEY access logging, 30
DBR invocation threshold, 192
DBR recovery duration, 193
DBR_DURATION, 193
DBR_RATIO, 192, 193
DECnet, 134, 135
Detected Asynchronous Pre-Fetch, 172

Emergency AIJ facility, 187
EXPORT/IMPORT, 199
Fast Commit, 78, 173, 179, 182, 193, 209, 229, 244
Fast Incremental Backup, 78
FILID, 78, 79
Full database backup threshold, 194
FULL_BACKUP_INTRVL, 194
GB IO-saved threshold, 175
GB pool hit threshold, 174
GB_IO_SAVED_RATIO, 175
GB_POOL_HIT_RATIO, 174
Hash duplicate fetch threshold, 208
Hash duplicate store threshold, 208, 210, 211
Hash table queue length threshold, 210
HASH_FETCH_DUP_RATIO, 208
HASH_STORE_DUP_RATIO, 208, 210, 211
help, 48, 49
Hot Standby, 45, 80, 85, 87, 164, 188, 189
LB/AS page hit threshold, 174
LB_PAGE_HIT_RATIO, 174
lock deadlock logging, 30
lock timeout logging, 30
Logical name, 11, 12
Markpoint, 33, 34, 35
MAX_HASH_QUE_LEN, 210
MAX_LOCK_STALL, 205
MAX_RO_TX_DURATION, 178
MAX_RO_TX_PERCENTILE, 178
MAX_RW_TX_DURATION, 177
MAX_TX_DURATION, 177
MAX_TX_PERCENTILE, 177
MEMBIT, 82
mixed format, 198
Network bandwidth threshold, 189
Notepad, 148
Nothing to report with current configuration, 169
Online Analysis, 16, 168, 169, 211
Online Analysis facility, 16, 46, 75, 76, 77, 80, 81, 83, 84, 85, 86, 87, 168, 169, 170, 171, 211
Oracle Trace, 211
Pages checked threshold, 199
PAGES_CHECKED_RATIO, 199
Process Monitoring facility, 14, 16, 36, 37, 39, 145
RCACHE, 84
RDM\$BIND_ABW_ENABLED, 173
RDM\$BIND_AIJ_ARB_COUNT, 184
RDM\$BIND_AIJ_EMERGENCY_DIR, 188
RDM\$BIND_AIJ_IO_MIN, 185
RDM\$BIND_ALS_CREATE_AIJ, 187, 188
RDM\$BIND_APF_ENABLED, 171
RDM\$BIND_BATCH_MAX, 190
RDM\$BIND_BUFOBJ_ENABLED, 253
RDM\$BIND_CLEAN_BUF_CNT, 190
RDM\$BIND_DAPF_ENABLED, 172
RDM\$BIND_RUJ_ALLOC_BLKCNT, 191

RDM\$BIND_RUJ_EXTEND_BLKCNT, 191
RDM\$BIND_STATS_AIJ_ARBS_PER_IO, 185
RDM\$BIND_STATS_AIJ_BKGRD_ARB_RATIO, 186
RDM\$BIND_STATS_AIJ_BLKs_PER_IO, 185
RDM\$BIND_STATS_AIJ_CACHE_OVF_RATIO, 187
RDM\$BIND_STATS_AIJ_NETWORK_BPS, 189
RDM\$BIND_STATS_AIJ_SEC_TO_EXTEND, 182
RDM\$BIND_STATS_BTR_FETCH_DUP_RATIO, 207
RDM\$BIND_STATS_BTR_LEF_FETCH_RATIO, 207
RDM\$BIND_STATS_BTR_STORE_DUP_RATIO, 207
RDM\$BIND_STATS_DBR_DURATION, 193
RDM\$BIND_STATS_DBR_RATIO, 192, 193
RDM\$BIND_STATS_DISABLED, 11, 12
RDM\$BIND_STATS_FULL_BACKUP_INTRVL, 194
RDM\$BIND_STATS_GB_IO_SAVED_RATIO, 175
RDM\$BIND_STATS_GB_POOL_HIT_RATIO, 174
RDM\$BIND_STATS_HASH_FETCH_DUP_RATIO, 208
RDM\$BIND_STATS_HASH_STORE_DUP_RATIO, 208, 210, 211
RDM\$BIND_STATS_LB_PAGE_HIT_RATIO, 174
RDM\$BIND_STATS_MAX_HASH_QUE_LEN, 210
RDM\$BIND_STATS_MAX_LOCK_STALL, 205
RDM\$BIND_STATS_MAX_RO_TX_DURATION, 178
RDM\$BIND_STATS_MAX_RO_TX_PERCENTILE, 178
RDM\$BIND_STATS_MAX_RW_TX_DURATION, 177
RDM\$BIND_STATS_MAX_RW_TX_PERCENTILE, 177
RDM\$BIND_STATS_MAX_TX_DURATION, 177
RDM\$BIND_STATS_MAX_TX_PERCENTILE, 177
RDM\$BIND_STATS_PAGES_CHECKED_RATIO, 199
RDM\$BIND_STATS_RECS_FETCHED_RATIO, 202
RDM\$BIND_STATS_RECS_STORED_RATIO, 201
RDM\$BIND_STATS_RUJ_EXTEND_RATIO, 191
RDM\$BIND_STATS_RUJ_SYNC_IO_RATIO, 191
RDM\$BIND_STATS_SPAM_PAG_FET_RATIO, 199
RDM\$BIND_STATS_SPAM_REC_STO_RATIO, 200
RDM\$BIND_STATS_VERB_SUCCESS_RATIO, 176
RDM\$BIND_WORK_FILE, 251
RDM\$BIND_WORK_VM, 251
Read/Write Transaction duration percentile, 177
Read/Write Transaction duration threshold, 178
Read-Only Transaction duration percentile, 178
Read-Only Transaction duration threshold, 178
Records fetched threshold, 202
Records stored threshold, 201
RECS_FETCHED_RATIO, 202
RECS_STORED_RATIO, 201
RMU Backup, 75, 199
RMU Backup After_Journal, 75
RMU Close, 5, 8, 12, 13
RMU Dump Header, 45, 92, 93, 95, 188, 222, 258
RMU Open, 8, 12, 13, 14, 36, 38, 54, 146
RMU Repair, 199
RMU Server, 144
RMUwin, 6
Rootfile, 75, 76, 77, 79, 81, 84, 86, 88, 139, 167, 229, 241, 244

Row Cache Server, 144, 150, 165
RTUPB, 86
RUJ extend threshold, 191
RUJ_EXTEND_RATIO, 191
RUJ_SYNC_IO_RATIO, 191
Seconds to AIJ extend/switch, 182
SEQBLK, 86
single-file database, 194
Single-Step, 33
Snapshot File, 62, 81, 209, 216, 229, 231, 232, 241, 242, 244
Space Area Management, 195
SPAM, 195
SPAM pages fetched threshold, 199
SPAM records stored threshold, 200
SPAM_PAG_FET_RATIO, 199
SPAM_REC_STO_RATIO, 200
stall message logging, 30
Storage Area File, 62, 229, 231, 232, 241, 244
Synchronous RUJ I/O threshold, 191
TCP/IP, 134, 135
TPS, 229
Transaction duration percentile, 177
Transaction duration threshold, 176, 177
TSNBLK, 88, 164
User-Defined Event, 151
VERB_SUCCESS_RATIO, 176
VLM, 88, 252, 257