

Autonomous Health Framework

User's Guide



18c
E83825-03
March, 2018

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Autonomous Health Framework User's Guide, 18c

E83825-03

Copyright © 2016, 2018, Oracle and/or its affiliates. All rights reserved.

Primary Author: Nirmal Kumar, Janet Stern

Contributing Authors: Richard Strohm, Mark Bauer, Douglas Williams, Aparna Kamath, Subhash Chandra

Contributors: Girdhari Ghantiyala, Gareth Chapman, Robert Caldwell, Vern Wagman, Mark Scardina, Ankita Khandelwal, Girish Adiga, Walter Battistella, Jesus Guillermo Munoz Nunez, Sahil Kumar, Daniel Semler, Carol Colrain

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xviii
Documentation Accessibility	xviii
Related Documentation	xviii
Conventions	xix

Changes in this Release for Autonomous Health Framework Users Guide 18c

New Features for Oracle ORAchk and Oracle EXAchk 12.1.0.2.7	xx
New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.1	xxiii
New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.2	xxvi
New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.3	xxvii
New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.4	xxix
New Features for Oracle Trace File Analyzer	xxxi
New Features for Oracle Cluster Health Advisor	xxxii
New Features for Oracle Database Quality of Service (QoS) Management	xxxii

1 Introduction to Oracle Autonomous Health Framework

1.1 Oracle Autonomous Health Framework Problem and Solution Space	1-1
1.1.1 Availability Issues	1-1
1.1.2 Performance Issues	1-3
1.2 Components of Oracle Autonomous Health Framework	1-4
1.2.1 Introduction to Oracle ORAchk and Oracle EXAchk	1-4
1.2.2 Introduction to Cluster Health Monitor	1-5
1.2.3 Introduction to Oracle Trace File Analyzer	1-6
1.2.4 Introduction to Oracle Cluster Health Advisor	1-6
1.2.5 Introduction to Memory Guard	1-7
1.2.6 Introduction to Hang Manager	1-8
1.2.7 Introduction to Oracle Database Quality of Service (QoS) Management	1-8

Part I Analyzing the Cluster Configuration

2 Analyzing Risks and Complying with Best Practices

2.1	Using Oracle ORAchk and Oracle EXAchk to Automatically Check for Risks and System Health	2-2
2.2	Email Notification and Health Check Report Overview	2-3
2.2.1	First Email Notification	2-3
2.2.2	What does the Health Check Report Contain?	2-3
2.2.3	Subsequent Email Notifications	2-5
2.3	Configuring Oracle ORAchk and Oracle EXAchk	2-6
2.3.1	Deciding Which User Should Run Oracle ORAchk or Oracle EXAchk	2-6
2.3.2	Handling of Root Passwords	2-8
2.3.3	Configuring Email Notification System	2-8
2.4	Using Oracle ORAchk and Oracle EXAchk to Manually Generate Health Check Reports	2-12
2.4.1	Running Health Checks On-Demand	2-12
2.4.2	Running Health Checks in Silent Mode	2-14
2.4.3	Running On-Demand With or Without the Daemon	2-15
2.4.4	Generating a Diff Report	2-16
2.4.5	Sending Results by Email	2-16
2.5	Managing the Oracle ORAchk and Oracle EXAchk Daemons	2-17
2.5.1	Starting and Stopping the Daemon	2-17
2.5.2	Configuring the Daemon for Automatic Restart	2-18
2.5.3	Setting and Getting Options for the Daemon	2-18
2.5.3.1	AUTORUN_SCHEDULE	2-19
2.5.3.2	AUTORUN_FLAGS	2-20
2.5.3.3	NOTIFICATION_EMAIL	2-20
2.5.3.4	collection_retention	2-21
2.5.3.5	PASSWORD_CHECK_INTERVAL	2-21
2.5.3.6	Setting Multiple Option Profiles for the Daemon	2-22
2.5.3.7	Getting Existing Options for the Daemon	2-23
2.5.4	Querying the Status and Next Planned Daemon Run	2-26
2.6	Tracking Support Incidents	2-27
2.7	Tracking File Attribute Changes and Comparing Snapshots	2-28
2.7.1	Using the File Attribute Check With the Daemon	2-29
2.7.2	Taking File Attribute Snapshots	2-29
2.7.3	Including Directories to Check	2-30
2.7.4	Excluding Directories from Checks	2-30
2.7.5	Rechecking Changes	2-31
2.7.6	Designating a Snapshot As a Baseline	2-31

2.7.7	Restricting System Checks	2-32
2.7.8	Removing Snapshots	2-32
2.8	Collecting and Consuming Health Check Data	2-32
2.8.1	Selectively Capturing Users During Login	2-33
2.8.2	Bulk Mapping Systems to Business Units	2-35
2.8.3	Adjusting or Disabling Old Collections Purging	2-36
2.8.4	Uploading Collections Automatically	2-38
2.8.5	Viewing and Reattempting Failed Uploads	2-40
2.8.6	Authoring User-Defined Checks	2-41
2.8.7	Finding Which Checks Require Privileged Users	2-44
2.8.8	Creating or Editing Incidents Tickets	2-45
2.8.8.1	Creating Incident Tickets	2-45
2.8.9	Viewing Clusterwide Linux Operating System Health Check (VMPScan)	2-46
2.9	Locking and Unlocking Storage Server Cells	2-47
2.10	Integrating Health Check Results with Other Tools	2-48
2.10.1	Integrating Health Check Results with Oracle Enterprise Manager	2-48
2.10.2	Integrating Health Check Results with Third-Party Tool	2-50
2.10.3	Integrating Health Check Results with Custom Application	2-51
2.11	Troubleshooting Oracle ORAchk and Oracle EXAchk	2-53
2.11.1	How to Troubleshoot Oracle ORAchk and Oracle EXAchk Issues	2-53
2.11.2	How to Capture Debug Output	2-54
2.11.3	Remote Login Problems	2-55
2.11.4	Permission Problems	2-56
2.11.5	Slow Performance, Skipped Checks and Timeouts	2-57

3 Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

3.1	Oracle Cluster Health Advisor Architecture	3-2
3.2	Monitoring the Oracle Real Application Clusters (Oracle RAC) Environment with Oracle Cluster Health Advisor	3-3
3.3	Using Cluster Health Advisor for Health Diagnosis	3-4
3.4	Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment	3-6
3.5	Viewing the Details for an Oracle Cluster Health Advisor Model	3-8
3.6	Managing the Oracle Cluster Health Advisor Repository	3-9
3.7	Viewing the Status of Cluster Health Advisor	3-10

Part II Automatically Monitoring the Cluster

4 Collecting Operating System Resources Metrics

4.1	Understanding Cluster Health Monitor Services	4-1
4.2	Collecting Cluster Health Monitor Data	4-2
4.3	Using Cluster Health Monitor from Enterprise Manager Cloud Control	4-2

5 Monitoring System Metrics for Cluster Nodes

5.1	Monitoring Oracle Clusterware with Oracle Enterprise Manager	5-1
5.2	Monitoring Oracle Clusterware with Cluster Health Monitor	5-3
5.3	Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures	5-3

Part III Monitoring and Managing Database Workload Performance

6 Introduction to Oracle Database QoS Management

6.1	What Is Oracle Database QoS Management?	6-1
6.2	Benefits of Using Oracle Database QoS Management	6-2
6.3	Overview of Oracle Database QoS Management	6-2
6.3.1	How Does Oracle Database QoS Management Work?	6-3
6.3.1.1	Oracle Database QoS Management and Server Pools	6-4
6.3.1.2	Oracle Database QoS Management and Instance Caging	6-4
6.3.1.3	Oracle Database QoS Management and Services	6-5
6.3.2	Overview of Policy Sets	6-6
6.3.3	Overview of Server Pools	6-7
6.3.4	How Server Pools Are Used by Oracle Database QoS Management	6-10
6.3.5	Overview of Performance Classes	6-11
6.3.5.1	Performance Class Tags	6-11
6.3.5.2	Applying Classifiers to Work Requests	6-12
6.3.5.3	Using Additional Filters for Classifying Work Requests	6-12
6.3.5.4	Deciding to Create New Performance Classes	6-13
6.3.6	Overview of Performance Policies and Performance Objectives	6-14
6.3.6.1	Overview of Performance Objectives	6-14
6.3.6.2	Overview of Server Pool Directive Overrides	6-15
6.3.6.3	Overview of Performance Class Ranks	6-17
6.3.6.4	Oracle Database QoS Management Policy Workload Criticality Determines Database Startup Order	6-18
6.3.7	How Oracle Database QoS Management Collects and Analyzes Performance Data	6-18
6.3.8	Overview of Recommendations	6-19
6.3.8.1	How Oracle Database QoS Management Generates Recommendations	6-20

6.3.8.2	Types of Recommendations	6-20
6.3.8.3	Choosing the Best Recommendation	6-22
6.3.8.4	Contents of a Recommendation	6-23
6.3.8.5	Overview of Implementing Recommendations	6-24
6.3.8.6	Example: How Recommendations Are Generated	6-25
6.4	What Does Oracle Database QoS Management Manage?	6-26
6.4.1	Managing Database Resources to Meet Service Levels	6-26
6.4.1.1	Database Services	6-27
6.4.1.2	Oracle Database Resource Manager	6-28
6.4.1.3	Oracle Clusterware	6-28
6.4.1.4	Run-time Connection Load Balancing	6-29
6.4.2	High Availability Management and Oracle Database QoS Management	6-29
6.5	Overview of Metrics	6-30
6.5.1	Performance Metrics	6-31
6.5.2	Resource Metrics	6-31
6.5.3	Performance Satisfaction Metrics	6-32
6.5.4	Using Metrics to Identify Performance Issues	6-32

7 Supported Workloads and Strategies

7.1	Supported Configurations for Oracle Database QoS Management	7-1
7.1.1	Supported Server Pool Configurations	7-2
7.1.2	Supported Database Configurations	7-2
7.1.3	Supported Service Configurations	7-3
7.1.4	Supported Workload and Objective Types	7-4
7.2	Strategies for Creating Classifiers for Performance Classes	7-5
7.3	Configuration Strategies for Effective Resource Management	7-6
7.3.1	About Resource Bottlenecks	7-6
7.3.2	CPU Resource Bottlenecks	7-6
7.3.3	Configuration Recommendations for Global Cache Resource Bottlenecks	7-7
7.3.4	Configuration Recommendations for I/O Resource Bottlenecks	7-7
7.3.5	Configuration Recommendations for Other Types of Bottlenecks	7-7
7.4	Sample Implementation of Oracle Database QoS Management	7-8
7.4.1	Description of the Demo System	7-8
7.4.2	Description of the System Workload	7-9
7.4.3	Initial Oracle Database QoS Management Configuration	7-10
7.5	Creating Oracle Database QoS Management Performance Policies for the Demo System	7-10
7.6	Managing Service Levels with Oracle Database QoS Management	7-11

8 Installing and Enabling Oracle Database QoS Management

8.1	Configuring Oracle Database QoS Management to Manage Oracle Database Workloads	8-1
8.1.1	Installing and Configuring Oracle Grid Infrastructure for a Cluster	8-2
8.1.2	Creating and Configuring Server Pools	8-2
8.1.3	Creating and Configuring an Oracle RAC Database	8-3
8.1.3.1	Modifying Database Initialization Parameters	8-4
8.1.3.2	Creating Database Services	8-4
8.1.4	Creating Oracle Database QoS Management Administrator Accounts	8-5
8.1.5	Enabling Oracle Database QoS Management	8-5
8.1.5.1	Enable Oracle QoS Management at the Database Level	8-6
8.1.5.2	Create an Initial Policy Set	8-7
8.1.5.3	Enable Oracle QoS Management at the Cluster Level	8-8
8.1.6	About Multi-CPU Binding on Solaris and Quality of Service Management	8-8

9 Administering the Oracle Database QoS Management System

9.1	Determining If Oracle Database QoS Management is Enabled	9-1
9.1.1	Checking the Enabled Status for a Database	9-2
9.1.2	Checking the Enabled Status for the Cluster	9-2
9.2	Monitoring Performance with Oracle Database QoS Management	9-2
9.3	Using the Oracle Database QoS Management Dashboard	9-3
9.3.1	Accessing the Oracle Database QoS Management Dashboard	9-4
9.3.2	Enabling Oracle Database QoS Management for a Cluster	9-6
9.3.3	Disabling Oracle Database QoS Management for a Cluster	9-6
9.3.4	Interpreting the Performance Overview Graphs	9-7
9.3.5	Viewing Recommendations	9-8
9.3.6	Viewing Recommendation Details	9-9
9.3.7	Implementing Recommendations	9-11
9.4	Administering the Policy Set	9-11
9.4.1	Editing a Policy Set	9-12
9.4.2	Adding Server Pools to a Policy Set	9-15
9.4.3	Modifying Server Pool Settings	9-15
9.4.4	Adding Database Services to a Policy Set	9-16
9.4.5	Updating a Policy Set to Include a New Database	9-17
9.5	Managing Performance Classes	9-17
9.5.1	Creating a Performance Class	9-18
9.5.2	Deleting a Performance Class	9-19
9.5.3	Renaming a Performance Class	9-20
9.5.4	Editing an Existing Performance Class	9-20

9.5.4.1	Adding Classifiers	9-21
9.5.4.2	Changing Classifiers	9-21
9.5.4.3	Deleting Classifiers	9-21
9.5.5	Specifying the Evaluation Order of the Classifiers	9-22
9.6	Managing Performance Policies	9-22
9.6.1	Creating a Performance Policy and Specifying Performance Objectives	9-22
9.6.2	Editing an Existing Performance Policy	9-24
9.6.3	Copying a Performance Policy	9-25
9.6.4	Setting the Current Performance Policy	9-26
9.6.4.1	Changing the Active Performance Policy from the Dashboard	9-26
9.6.4.2	Changing the Active Performance Policy from the Policy Set Editor Wizard	9-26
9.6.4.3	Changing the Active Performance Policy using a Script	9-26
9.6.5	Deleting a Performance Policy	9-27
9.6.6	Automatically Implementing Recommendations for a Performance Policy	9-27
9.6.7	Setting Server Pool Directive Overrides	9-27
9.7	Reviewing Performance Metrics	9-29
9.7.1	Viewing Performance Metrics for All Performance Classes	9-29
9.7.2	Viewing Performance Metrics for Individual Performance Classes	9-30
9.7.3	Configuring Alerts for Quality of Service Management Events	9-31
9.7.4	Viewing the Resource Wait Times Breakdown	9-32
9.8	Creating Administrative Users for Oracle Database QoS Management	9-33
9.8.1	QOSCTL Utility Reference	9-34
9.9	Editing the Resource Plan for Oracle Database QoS Management	9-35

10 Troubleshooting Oracle Database QoS Management

10.1	Common Problems	10-1
10.1.1	Cannot Enable Oracle Database Quality of Service Management	10-1
10.1.2	Cannot Enable Oracle Database QoS Management for a Database	10-2
10.1.3	Oracle Database Resource Manager Not Enabled and Resource Plan Errors	10-2
10.1.4	Do Not Have Access to a Server Pool	10-3
10.1.5	Server Pool Is Marked As Unmanageable	10-3
10.1.6	Metrics Are Missing For a Performance Class	10-4
10.1.7	Oracle Database QoS Management is not Generating Recommendations	10-4
10.1.8	Recently Added Server was Placed in the Wrong Server Pool	10-4
10.1.9	RMI Port Conflict Detected	10-5
10.2	Locating Log or Trace Files	10-5

Part IV Automatic Problem Solving

11 Resolving Memory Stress

11.1	Overview of Memory Guard	11-1
11.2	Memory Guard Architecture	11-2
11.3	Enabling Memory Guard in Oracle Real Application Clusters (Oracle RAC) Environment	11-3
11.4	Use of Memory Guard in Oracle Real Application Clusters (Oracle RAC) Deployment	11-4

12 Resolving Database and Database Instance Hangs

12.1	Hang Manager Architecture	12-1
12.2	Optional Configuration for Hang Manager	12-3
12.3	Hang Manager Diagnostics and Logging	12-4

Part V Collecting Diagnostic Data and Triaging, Diagnosing, and Resolving Issues

13 Oracle Trace File Analyzer

14 Getting Started with Oracle Trace File Analyzer

14.1	Supported Environments	14-1
14.2	Installing Oracle Trace File Analyzer on Linux or UNIX as root User in Daemon Mode	14-2
14.3	Installing Oracle Trace File Analyzer on Linux or UNIX as Non-root User in Non-Daemon Mode	14-3
14.4	Installing Oracle Trace File Analyzer on Microsoft Windows	14-3
14.5	Installing Oracle Trace File Analyzer on Microsoft Windows in Non-Daemon Mode	14-4
14.6	Oracle Trace File Analyzer Key Directories	14-4
14.7	Oracle Trace File Analyzer Command Interfaces	14-5
14.8	Masking Sensitive Data	14-5
14.9	Securing Access to Oracle Trace File Analyzer	14-6
14.10	Uninstalling Oracle Trace File Analyzer	14-7

15	Automatic Diagnostic Collections	
	<hr/>	
15.1	Collecting Diagnostics Automatically	15-1
15.2	Configuring Email Notification Details	15-2
16	On-demand Analysis and Diagnostic Collection	
	<hr/>	
16.1	Collecting Diagnostics and Analyzing Logs On-Demand	16-1
16.2	Viewing System and Cluster Summary	16-2
16.3	Investigating Logs for Errors	16-2
16.4	Analyzing Logs Using the Included Tools	16-4
16.5	Collecting Diagnostic Data and Using One Command Service Request Data Collections	16-6
16.6	Uploading Collections to Oracle Support	16-10
16.7	Changing Oracle Clusterware Trace Levels	16-10
16.7.1	tfactl dbglevel	16-11
17	Maintaining Oracle Trace File Analyzer to the Latest Version	
	<hr/>	
18	Performing Custom Collections	
	<hr/>	
18.1	Adjusting the Diagnostic Data Collection Period	18-1
18.2	Collecting from Specific Nodes	18-2
18.3	Collecting from Specific Components	18-2
18.4	Collecting from Specific Directories	18-3
18.5	Changing the Collection Name	18-4
18.6	Preventing Copying Zip Files and Trimming Files	18-5
18.7	Performing Silent Collection	18-6
18.8	Preventing Collecting Core Files	18-6
18.9	Collecting Incident Packaging Service (IPS) Packages	18-6
19	Managing and Configuring Oracle Trace File Analyzer	
	<hr/>	
19.1	Querying Oracle Trace File Analyzer Status and Configuration	19-1
19.2	Managing the Oracle Trace File Analyzer Daemon	19-3
19.3	Managing the Repository	19-3
19.3.1	Purging the Repository Automatically	19-4
19.3.2	Purging the Repository Manually	19-5
19.4	Managing Collections	19-5
19.4.1	Including Directories	19-5

19.4.2	Managing the Size of Collections	19-6
19.5	Configuring the Host	19-7
19.6	Configuring the Ports	19-7
19.7	Configuring SSL and SSL Certificates	19-8
19.7.1	Configuring SSL/TLS Protocols	19-8
19.7.2	Configuring Self-Signed Certificates	19-9
19.7.3	Configuring CA-Signed Certificates	19-10
19.7.4	Configuring SSL Cipher Suite	19-11

20 Oracle Trace File Analyzer Service

20.1	Overview of Oracle Trace File Analyzer Service	20-1
20.2	Cluster or Host View	20-2
20.3	Detailed View	20-3
20.4	File Browser	20-4
20.5	File Viewer	20-5
20.6	Merged Files	20-5
20.7	Importing CA Certificates into Tomcat	20-6

21 Managing Oracle Database and Oracle Grid Infrastructure Diagnostic Data

21.1	Managing Automatic Diagnostic Repository Log and Trace Files	21-1
21.2	Managing Disk Usage Snapshots	21-2
21.3	Purging Oracle Trace File Analyzer Logs Automatically	21-2

22 Troubleshooting Oracle Trace File Analyzer

22.1	Cluster Nodes are Not Showing As One Cluster When Viewed by Running the tfactl status Command	22-1
22.2	Oracle Trace File Analyzer is Not Starting and the init.tfa script is Missing After Reboot	22-2
22.3	Error Message Similar to "Can't locate **** in @inc (@inc contains:....)"	22-2
22.4	Non-Release Update Revisions (RURs) Oracle Trace File Analyzer Patching Fails on Remote Nodes	22-3
22.5	Non-Root Access is Not Enabled After Installation	22-3
22.6	TFA_HOME and Repository Locations are Moved After Patching or Upgrade	22-4
22.7	Oracle Trace File Analyzer Fails with TFA-00103 After Applying the July 2015 Release Update Revision (RUR) or Later	22-4
22.8	OSWatcher Parameters are Different After a Reboot or Otherwise Unexpectedly Different	22-10
22.9	Oracle Trace File Analyzer Installation or Oracle Trace File Analyzer Discovery (tfactl rediscover) Fails on Linux 7	22-11

22.10	OSWatcher Analyzer Fails When OSWatcher is Not Running from the TFA_HOME	22-12
22.11	Oracle Trace File Analyzer Fails to Start with com.sleepycat.je.EnvironmentLockedException Java Exception	22-12
22.12	Oracle Trace File Analyzer Startup Fails When Solution-Soft Time Machine Software is Installed, but Not Running on the System	22-13
22.13	Non-privileged User is Not Able to Run tfactl Commands?	22-13
22.14	Oracle Trace File Analyzer Daemon is Not Starting or Not Running?	22-14

Part VI Appendixes

Glossary

Index

List of Figures

2-1	Oracle Health Check Collections Manager - Administration	2-9
2-2	Oracle Health Check Collections Manager - Configure Email Server	2-9
2-3	Oracle Health Check Collections Manager - Notification Job Run status details	2-10
2-4	Oracle Health Check Collections Manager - Manage Notifications	2-10
2-5	Oracle Health Check Collections Manager - Sample Email Notification	2-11
2-6	Oracle Health Check Collections Manager - Sample Diff Report	2-12
2-7	Incidents Tab	2-28
2-8	Manage Users, User Roles and assign System to users	2-34
2-9	Don't Capture User Details (When Login)	2-34
2-10	Capture User Details (When Login)	2-35
2-11	Assign System to Business Unit	2-35
2-12	Bulk Mapping	2-36
2-13	Upload a mapping XML	2-36
2-14	Manage Email Server and Job Details	2-37
2-15	Configure Purging	2-37
2-16	User-Defined Checks Tab	2-41
2-17	User-Defined Checks Tab - Audit Check Type	2-41
2-18	User-Defined Checks Tab - Audit Check Type - OS Check	2-42
2-19	User-Defined Checks Tab - Available Audit Checks	2-43
2-20	User-Defined Checks Tab - Download User-Defined Checks	2-44
2-21	Oracle ORAchk - Privileged User	2-45
2-22	Clusterwide Linux Operating System Health Check (VMPScan)	2-47
2-23	Compliance Dashboard	2-49
2-24	Compliance Standards	2-49
2-25	Compliance Standards Drill-Down	2-49
2-26	Third-Party Tool Integration	2-50
2-27	Skipped Checks	2-57
3-1	Oracle Cluster Health Advisor Architecture	3-2
3-2	Cluster Health Advisor Diagnosis HTML Output	3-5
4-1	EMCC - Cluster Health Monitoring	4-3
4-2	Cluster Health Monitoring - Real Time Data	4-4
4-3	Cluster Health Monitoring - Historical Data	4-4
6-1	Instance Caging and CPU Slices	6-5
6-2	Elements of an Oracle Database QoS Management Policy Set	6-7
6-3	Diagram of Server Pools, Oracle Databases, and Database Services	6-8

6-4	Server Pools and CPU Slices	6-9
6-5	Baseline Resource Management by Performance Policy	6-16
6-6	Diagram of Oracle Database QoS Management Server Architecture	6-19
6-7	Example of the Analysis for a Recommended Action	6-24
7-1	Illustration of a Sample Workload	7-9
9-1	Performance Satisfaction Metrics for Measure-Only Performance Classes	9-3
9-2	Oracle Database Quality of Service Management Dashboard	9-6
9-3	Example of the Performance Overview Graphs	9-7
9-4	Performance Classes Not Meeting Their Performance Objectives	9-9
9-5	Resource Wait Time and Resource Usage Time Charts for a Performance Class	9-31
11-1	Memory Guard Architecture	11-2
12-1	Hang Manager Architecture	12-2
15-1	Automatic Diagnostic Collections	15-1
16-1	On-Demand Collections	16-2

List of Tables

2-1	AUTORUN_SCHEDULE	2-19
2-2	AUTORUN_FLAGS	2-20
2-3	Uploading Collection Results into a Database	2-51
2-4	Timeout Controlling	2-58
9-1	Summary of Commands for the QOSCTL Utility	9-34
14-1	Key Oracle Trace File Analyzer Directories	14-4
14-2	Oracle Trace File Interfaces	14-5
15-1	Log Entries that Trigger Automatic collection	15-2
16-1	Tools included in Linux and UNIX	16-4
16-2	Tools included in Microsoft Windows	16-5
16-3	One Command Service Request Data Collections	16-7
16-4	SRDC collections	16-8
16-5	tfactl dbglevel Command Parameters	16-11
18-1	Ways to Specify the Collection Period	18-1
18-2	Component Options	18-3
18-3	tfactl ips Command Parameters	18-6
19-1	Configuration Listing and Descriptions	19-1
1	Generic Commands	4
2	Scope of Checks	5
3	Managing Output	6
4	Uploading Results to Database	7
5	Daemon Options	9
6	List of Oracle ORAchk and Oracle EXAchk File Attribute Tracking Options	10
7	oclumon debug Command Parameters	11
8	oclumon dumpnodeview Command Parameters	13
9	oclumon dumpnodeview SYSTEM View Metric Descriptions	14
10	oclumon dumpnodeview PROCESSES View Metric Descriptions	17
11	oclumon dumpnodeview DEVICES View Metric Descriptions	18
12	oclumon dumpnodeview NICS View Metric Descriptions	18
13	oclumon dumpnodeview FILESYSTEMS View Metric Descriptions	19
14	oclumon dumpnodeview PROTOCOL ERRORS View Metric Descriptions	19
15	oclumon dumpnodeview CPUS View Metric Descriptions	20
16	oclumon manage Command Parameters	23
17	crsctl query calog Command Parameters	26
18	Cluster Resource Activity Log Fields	27

19	chactl monitor Command Parameters	35
20	Basic TFACTL commands	49
21	tfactl diagnosetfa Command Parameters	50
22	tfactl set Command Parameters	51
23	tfactl access Command Parameters	53
24	tfactl analyze Command Parameters	59
25	tfactl analyze -type Parameter Arguments	60
26	tfactl run Command Parameters	62
27	tfactl run Analysis Tools Parameters	62
28	tfactl run Profiling Tools Parameters	62
29	tfactl toolstatus Output	63
30	tfactl directory Command Parameters	68
31	tfactl ips Command Parameters	70
32	tfactl ips ADD Command Parameters	73
33	tfactl ips ADD FILE Command Parameters	73
34	tfactl ips COPY IN FILE Command Parameters	74
35	tfactl ips REMOVE Command Parameters	74
36	tfactl ips ADD NEW INCIDENTS PACKAGE Command Parameters	75
37	tfactl ips GET REMOTE KEYS FILE Command Parameters	75
38	tfactl ips CREATE PACKAGE Command Parameters	76
39	tfactl ips GENERATE PACKAGE Command Parameters	77
40	tfactl ips DELETE PACKAGE Command Parameters	78
41	tfactl ips GET MANIFEST FROM FILE Command Parameters	79
42	tfactl ips PACK Command Parameters	79
43	tfactl ips SET CONFIGURATION Command Parameters	81
44	tfactl print Command Parameters	83
45	tfactl managelogs Purge Options	86
46	tfactl managelogs Show Options	86

Preface

Oracle Autonomous Health Framework User's Guide explains how to use the Oracle Autonomous Health Framework diagnostic components.

The diagnostic components include Oracle ORAchk, Oracle EXAchk, Cluster Health Monitor, Oracle Trace File Analyzer Collector, Oracle Cluster Health Advisor, Memory Guard, and Hang Manager.

Oracle Autonomous Health Framework User's Guide also explains how to install and configure Oracle Trace File Analyzer Collector.

This Preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documentation](#)
- [Conventions](#)

Audience

Database administrators can use this guide to understand how to use the Oracle Autonomous Health Framework diagnostic components. This guide assumes that you are familiar with Oracle Database concepts.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documentation

For more information, see the following Oracle resources:

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database 2 Day DBA*
- *Oracle Database Concepts*
- *Oracle Database Examples Installation Guide*
- *Oracle Database Licensing Information*
- *Oracle Database Release Notes*
- *Oracle Database Upgrade Guide*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Real Application Clusters Installation Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Changes in this Release for Autonomous Health Framework Users Guide 18c

This preface lists changes in Oracle Autonomous Health Framework for Oracle Database 18c.

- [New Features for Oracle ORAchk and Oracle EXAchk 12.1.0.2.7](#)
- [New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.1](#)
- [New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.2](#)
- [New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.3](#)
- [New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.4](#)
- [New Features for Oracle Trace File Analyzer](#)
- [New Features for Oracle Cluster Health Advisor](#)
- [New Features for Oracle Database Quality of Service \(QoS\) Management](#)

New Features for Oracle ORAchk and Oracle EXAchk 12.1.0.2.7

These are new features for Oracle ORAchk and Oracle EXAchk 12.1.0.2.7.

- [Simplified Enterprise-Wide Data Configuration and Maintenance](#)
- [Tracking Changes to File Attributes](#)
- [Find Health Checks that Require Privileged Users to Run](#)
- [Support for Broader Range of Oracle Products](#)
- [Easier to Run Oracle EXAchk on Oracle Exadata Storage Servers](#)
- [New Health Checks for Oracle ORAchk and Oracle EXAchk](#)

Simplified Enterprise-Wide Data Configuration and Maintenance

This release contains several changes that simplify system and user configuration and maintenance, and upload health check collection results.

- [Bulk Mapping Systems to Business Units](#)
- [Selectively Capturing Users During Log In](#)
- [Configuring Details for Upload of Health Check Collection Results](#)
- [Viewing and Reattempting Failed Uploads](#)

- [Managing Oracle Health Check Collection Purges](#)

Bulk Mapping Systems to Business Units

Oracle Health Check Collections Manager provides an XML bulk upload option so that you can quickly map many systems to business units.

Related Topics

- [Bulk Mapping Systems to Business Units](#)
Oracle Health Check Collections Manager provides an XML bulk upload option so that you can quickly map many systems to business units.

Selectively Capturing Users During Log In

By default, Oracle Health Check Collections Manager captures details of the users who log in using LDAP authentication and assigns Oracle Health Check Collections Manager role to the users.

Related Topics

- [Selectively Capturing Users During Login](#)
Configure Oracle Health Check Collections Manager to capture user details and assign the users Oracle Health Check Collections Manager roles.

Configuring Details for Upload of Health Check Collection Results

Configure Oracle ORAchk and Oracle EXAchk to automatically upload health check collection results to the Oracle Health Check Collections Manager database.

Related Topics

- [Uploading Collections Automatically](#)
Configure Oracle ORAchk and Oracle EXAchk to upload check results automatically to the Oracle Health Check Collections Manager database.

Viewing and Reattempting Failed Uploads

Use the new option `-checkfaileduploads` to find failed uploads.

Related Topics

- [Viewing and Reattempting Failed Uploads](#)
Configure Oracle ORAchk and Oracle EXAchk to display and reattempt to upload the failed uploads.

Managing Oracle Health Check Collection Purges

Oracle Health Check Collections Manager now by default purges collections that are older than three months.

Related Topics

- [Adjusting or Disabling Old Collections Purging](#)
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.

Tracking Changes to File Attributes

Use the Oracle ORAchk and Oracle EXAchk option `-fileattr` to track changes to file attributes.

If run with the `-fileattr` option, then Oracle ORAchk and Oracle EXAchk search all files within Oracle Grid Infrastructure and Oracle Database homes by default. Also, specify the list of directories, subdirectories, and files to monitor, and then compare snapshots for changes.

Related Topics

- [Tracking File Attribute Changes](#)
Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.
- [Tracking File Attribute Changes and Comparing Snapshots](#)
Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.

Find Health Checks that Require Privileged Users to Run

Use the new privileged user filter to identify health checks that require a privileged user to run.

Related Topics

- [Finding Which Checks Require Privileged Users](#)
Use the **Privileged User** filter in the Health Check Catalogs to find health checks that must be run by privileged users, such as `root`.

Support for Broader Range of Oracle Products

Health check support has been broadened to include Linux operating system health checks (Oracle ORAchk only), External ZFS Storage Appliance health checks (Oracle EXAchk on Exalogic only), and Oracle Enterprise Manager Cloud Control 13.1.

Related Topics

- [Viewing Clusterwide Linux Operating System Health Check \(VMPScan\)](#)
On Linux systems, view a summary of the VMPScan report in the Clusterwide Linux Operating System Health Check (VMPScan) section of the Health Check report.

Easier to Run Oracle EXAchk on Oracle Exadata Storage Servers

Run Oracle EXAchk from Oracle Exadata storage servers without SSH connectivity from the database server to the storage server.

Related Topics

- [Locking and Unlocking Storage Server Cells](#)
Beginning with version 12.1.0.2.7, use Oracle EXAchk to lock and unlock storage server cells.

New Health Checks for Oracle ORAchk and Oracle EXAchk

New health checks have been included for:

- Oracle Exadata
- Oracle SuperCluster
- Oracle Exalogic
- Oracle ZFS Storage
- Oracle Enterprise Linux
- Oracle Solaris Cluster
- Oracle Database and Oracle Grid Infrastructure
- Oracle Enterprise Manager Cloud Control Oracle Management Service (OMS) and Repository

Refer to My Oracle Support note 1268927.2, click the **Health Check Catalog** tab to download and view the list of health checks.

Related Topics

- <https://support.oracle.com/rs?type=doc&id=1268927.2>

New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.1

These are new features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.1.

- [Easier to Use and Upgrade Oracle Health Check Collections Manager](#)
Oracle Health Check Collections Manager has been enhanced with a new easier to navigate interface based on the Oracle Application Express (APEX) 5 theme.
- [More Flexibility for Running Root Privileged Checks](#)
Ability to change the location of script used for running root checks, a different location from other temp files.
- [Oracle ORAchk Specific New Features](#)
Beginning with release 12.2.0.1.1, Oracle ORAchk adds support for the latest version of IBM AIX 7.2 and Oracle Database Appliances Small and Medium.
- [Oracle EXAchk Specific New Features](#)
Beginning with release 12.2.0.1.1, Oracle EXAchk adds support for the newest Oracle Engineered Systems, Role Based Access Control (RBAC) for Oracle EXAchk on SuperCluster, and send email from storage cells.

Easier to Use and Upgrade Oracle Health Check Collections Manager

Oracle Health Check Collections Manager has been enhanced with a new easier to navigate interface based on the Oracle Application Express (APEX) 5 theme.

This new interface is available only if you have APEX 5 installed on your database. Oracle ORAchk and Oracle EXAchk continue to include the APEX 4 based Oracle Health Check Collections Manager application. However, all enhancements in new releases go into the APEX 5 Oracle Health Check Collections Manager application.

Upgrade Oracle Health Check Collections Manager from Oracle ORAchk or Oracle EXAchk. Oracle Health Check Collections Manager upgrades to the latest version of whichever application your database supports. You get the new theme interface only if you have APEX 5.

If the Oracle Health Check Collections Manager schema changes in the future and if Oracle ORAchk and Oracle EXAchk require Oracle Health Check Collections Manager upgrade, then Oracle ORAchk and Oracle EXAchk prompt you to upgrade Oracle Health Check Collections Manager.

During upgrade, Oracle Health Check Collections Manager goes offline and is not available to receive new collections. If any collection fails to upload during upgrade, then you can upload them again by viewing and reattempting to upload the failed uploads.

If you are installing Oracle Health Check Collections Manager for the first time, then install through the APEX workspace. Use whichever Oracle Health Check Collections Manager `sql` script is applicable for the APEX version you are using:

- **APEX 4.x: CollectionManager_App.sql**
- **APEX 5.x: Apex5_CollectionManager_App.sql**

You can now sort all collection reports by any column no matter what APEX version you are using, making it easier to analyze results.

The Purge job feature has been improved to run daily rather than every three months to remove data older than three months.

Bulk mapping of business units is now easier as Oracle Health Check Collections Manager automatically creates any new business units found in the uploaded XML mapping file.

More Flexibility for Running Root Privileged Checks

Ability to change the location of script used for running root checks, a different location from other temp files.

A large number of the Oracle stack health checks do not require `root` access. However, a subset of health checks need to run with `root` privileges.

The script used to run these root privilege checks is `root_orachk.sh` or `root_exachk.sh` depending on the tool you are using. By default, this root script is created in the temporary directory, that is, `$HOME`. You can change the location by setting the `RAT_TMPDIR` environment variable.

You can choose to have the root script created outside of the standard temporary directory in a custom directory by setting the `RAT_ROOT_SH_DIR` environment variable:

```
export RAT_ROOT_SH_DIR=/orahome/oradb/
```

You can use a specific location for `sudo` remote access as follows:

```
export RAT_ROOT_SH_DIR=/mylocation
```

Then add an entry in the `/etc/sudoers` file as follows:

```
oracle ALL=(root) NOPASSWD:/mylocation/root_orachk.sh
```

Oracle ORAchk Specific New Features

Beginning with release 12.2.0.1.1, Oracle ORAchk adds support for the latest version of IBM AIX 7.2 and Oracle Database Appliances Small and Medium.

- **Support for AIX 7.2**

Oracle ORAchk adds support for the latest version of IBM AIX 7.2 in addition to the existing support for versions 5.2, 5.3, 6.1, and 7.1.

- **Support for New Oracle Database Appliances**

Oracle ORAchk adds support for new Oracle Database Appliances Small and Medium in addition to the existing support for Oracle Database Appliance HA.

Oracle EXAchk Specific New Features

Beginning with release 12.2.0.1.1, Oracle EXAchk adds support for the newest Oracle Engineered Systems, Role Based Access Control (RBAC) for Oracle EXAchk on SuperCluster, and send email from storage cells.

- **Support for Newest Oracle Engineered Systems**

Oracle EXAchk extends support to include some of the newest Oracle Engineered Systems and their software:

- Oracle MiniCluster – The latest addition to the Oracle Super Cluster family of Engineered Systems
- Oracle Exalytics Software release 2.0 – The latest software release for the Oracle Exalytics In-Memory Machine

- **Oracle EXAchk on SuperCluster - Support for Role Based Access Control**

When running Oracle EXAchk on Oracle SuperCluster you can use Role Based Access Control (RBAC) to run `root` privileged checks without requiring a root user.

Ensure that you run the root checks as a user with a root equivalent access role.

- **Oracle EXAchk on Exadata – Send Email When Unable to Send from Storage Cells**

If the Oracle Exadata storage cells are unable to send email, then you can use the `-usecompute` option to send the email from the database server instead.

For example:

```
./exachk -testemail
"NOTIFICATION_EMAIL=some.person@example.com,another.person@example.com" -
usecompute
```

or

```
./exachk -sendemail  
"NOTIFICATION_EMAIL=some.person@example.com,another.person@example.com" -  
usecompute
```

New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.2

These are new features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.2.

- [Enhanced Elastic Stack Integration](#)
The existing Oracle ORAchk and Oracle EXAchk integration with the Elastic Stack (previously called ELK stack) has been enhanced.
- [Oracle Health Check Collections Manager Health Score Filter](#)
A new filter, **Health Score** has been added to Oracle Health Check Collections Manager.
- [Support for Automated Platinum Service Requests for Qualified OEM Fault Events](#)
Oracle EXAchk will now check the Oracle Platinum Services monitoring agent, the monitoring user, monitored databases, and their diagnostic directories.

Enhanced Elastic Stack Integration

The existing Oracle ORAchk and Oracle EXAchk integration with the Elastic Stack (previously called ELK stack) has been enhanced.

Many new field tags have been added to the Oracle ORAchk and Oracle EXAchk JSON output.

The extra tags allow dashboard filtering based on facts such as:

- Engineered System type
- Engineered System version
- Hardware type
- Node name
- Operating system version
- Rack identifier
- Rack type
- Database version

Use the Kibana dashboard to view health check compliance across the data center.

Now with these new field tags, you can also filter the results based on any combination of exposed system attributes.

Oracle Health Check Collections Manager Health Score Filter

A new filter, **Health Score** has been added to Oracle Health Check Collections Manager.

The **Health Score** filter enables you to filter the results to only show collections where the health score is less than or equal to the value specified.

The default value of the filter is 100, and therefore Oracle Health Check Collections Manager displays all the collections initially.

In addition, a number of other existing Oracle Health Check Collections Manager field labels have been renamed for ease of use.

Support for Automated Platinum Service Requests for Qualified OEM Fault Events

Oracle EXAchk will now check the Oracle Platinum Services monitoring agent, the monitoring user, monitored databases, and their diagnostic directories.

This helps to ensure quality of Oracle Platinum Services monitoring and that relevant diagnostics can be supplied for faster fault resolution.

New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.3

These are new features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.3.

- [Upgrade to Oracle Database 12.2 with Confidence](#)
This release includes lots of new checks to help you when upgrading Oracle Database to 12.2.
- [Improve Performance of SQL Queries](#)
Many new checks focus on the known issues in 12c Optimizer as well as SQL Plan Management.
- [Customize Oracle Health Check Collections Manager Email Notifications and Sample Data Usage](#)
Customize Oracle Health Check Collections Manager email notification based on one or more business units.
- [Prevent Oracle Clusterware Related Outages and Corruptions](#)
New checks to verify network configuration and operating system patching.
- [Guard Against Exadata Critical Issue EX35](#)
New checks detect an Exadata critical issue, which can cause node eviction.
- [Avoid an Oracle SuperCluster Critical Issue and Storage Problems](#)
New checks are now available for a critical issue and storage problems affecting Oracle SuperCluster.

Upgrade to Oracle Database 12.2 with Confidence

This release includes lots of new checks to help you when upgrading Oracle Database to 12.2.

New Checks include both pre- and post-upgrade verification to prevent problems related to:

- Operating system configuration
- Oracle Grid Infrastructure and Oracle Database patch prerequisites
- Oracle Database configuration

- Cluster configuration

Improve Performance of SQL Queries

Many new checks focus on the known issues in 12c Optimizer as well as SQL Plan Management.

The new checks target the problems such as:

- Wrong results returned
- High memory and CPU usage
- Errors such as `ORA-00600` or `ORA-07445`
- Issues with cursor usage
- Other general SQL plan management problems

Customize Oracle Health Check Collections Manager Email Notifications and Sample Data Usage

Customize Oracle Health Check Collections Manager email notification based on one or more business units.

By default, existing users who have already configured notification will receive notification for all business units.

It is now possible to delete or reload the sample data that Oracle Health Check Collections Manager ships with.

Prevent Oracle Clusterware Related Outages and Corruptions

New checks to verify network configuration and operating system patching.

The Oracle Cluster Registry subnet records can become stale due to configuration changes. This can lead to various startup problems.

Additionally, vulnerability to certain operating system bugs can lead to corruption in clustered databases.

Guard Against Exadata Critical Issue EX35

New checks detect an Exadata critical issue, which can cause node eviction.

In addition, various other Exadata best practice recommendations have been added.

Avoid an Oracle SuperCluster Critical Issue and Storage Problems

New checks are now available for a critical issue and storage problems affecting Oracle SuperCluster.

These detect vulnerability to a critical issue affecting Oracle SuperCluster. As well as checks to prevent database memory corruption and storage related problems.

New Features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.4

These are new features for Oracle ORAchk and Oracle EXAchk 12.2.0.1.4.

- [Three Times Faster Performance](#)
Oracle ORAchk and Oracle EXAchk have been rewritten with a focus on performance and extensibility.
- [Option to Run Only the Failed Checks](#)
New option allows Oracle ORAchk and Oracle EXAchk to run only the checks that failed previously.
- [Microsoft Windows Support Without Requiring Cygwin](#)
Since Oracle ORAchk has been rewritten in Python, Cygwin on Microsoft Windows is no more a requirement.
- [Run Multiple Instances at the Same Time](#)
It is now possible to run multiple instances of Oracle ORAchk and Oracle EXAchk at the same time on the same machine.
- [Oracle ORAchk Support for Oracle Grid Infrastructure with no Database](#)
Oracle ORAchk now supports Oracle Grid Infrastructure stand-alone checks where no database is installed.
- [New Oracle Stack Health Checks](#)
This release of Oracle ORAchk and Oracle EXAchk includes many new Oracle Stack Health Checks.

Three Times Faster Performance

Oracle ORAchk and Oracle EXAchk have been rewritten with a focus on performance and extensibility.

Internal testing results show that this version runs up to 3 times faster than 12.2.0.1.3. Performance has improved in low-resource environments.

Option to Run Only the Failed Checks

New option allows Oracle ORAchk and Oracle EXAchk to run only the checks that failed previously.

To run only the checks that failed previously:

- Generate a health check report
- Fix the issues identified
- Generate another health check report verifying only the issues that failed before

Use the failed checks option by passing in the HTML report, zip, or directory.

```
-failedchecks previous_result
```

Microsoft Windows Support Without Requiring Cygwin

Since Oracle ORAchk has been rewritten in Python, Cygwin on Microsoft Windows is no more a requirement.

Oracle ORAchk needs Python 2.7 software and libraries. The runtime provides details of what exactly the libraries it needs.

Run Multiple Instances at the Same Time

It is now possible to run multiple instances of Oracle ORAchk and Oracle EXAchk at the same time on the same machine.

As a result of this change the location of the `root_orachk.sh` and `root_exachk.sh` files have now changed. By default, they are created in the user's home directory.

If you run as non-root, then change this location in the `/etc/sudoers` file, for example,

Previous `/etc/sudoers` entry:

```
oracle ALL=(root) NOPASSWD:/home/oracle/.orachk/root_orachk.sh
```

New `/etc/sudoers` entry:

```
oracle ALL=(root) NOPASSWD:/home/oracle/root_orachk.sh
```

You can also change the location of those scripts by setting the environment variable `RAT_ROOT_SH_DIR`.

```
export RAT_ROOT_SH_DIR=/some/other/location
```

Oracle ORAchk Support for Oracle Grid Infrastructure with no Database

Oracle ORAchk now supports Oracle Grid Infrastructure stand-alone checks where no database is installed.

To run Oracle Grid Infrastructure checks in an environment with no Oracle Database has been installed, use the option:

```
-nordbms
```

For example:

```
$ ./orachk -nordbms
```

```
$ ./exachk -nordbms
```

New Oracle Stack Health Checks

This release of Oracle ORAchk and Oracle EXAchk includes many new Oracle Stack Health Checks.

- [Oracle ORAchk Health Check Catalog](#)

- [Oracle EXAchk Health Check Catalog](#)

The Health Check Catalog is also contained within the download.

New Features for Oracle Trace File Analyzer

Oracle Trace File Analyzer includes the following new features:

- **Oracle Trace File Analyzer Service**
Oracle Trace File Analyzer service helps DBAs and system administrators proactively monitor trace files across multiple member clusters on a centralized machine.
- **One Command Service Request Data Collections (SRDC)**
Oracle Trace File Analyzer can run SRDC collections with a single command.

Related Topics

- [Oracle Trace File Analyzer Service](#)
Oracle Trace File Analyzer Service is deployed as part of Domain Service Cluster (DSC) setup.
- [Collecting Diagnostic Data and Using One Command Service Request Data Collections](#)

New Features for Oracle Cluster Health Advisor

Oracle Cluster Health Advisor includes the following new features:

- **Oracle Cluster Health Advisor Cross-Database Analysis Support**
In consolidated and DBaaS private cloud deployments multiple databases share the same physical server and its resources. In its previous release, Oracle Cluster Health Advisor analyzed each hosted database instance individually.

Oracle Cluster Health Advisor could only detect whether the cause of a performance or availability issue was within itself or external. With the new cross-database analysis support, Oracle Cluster Health Advisor can target external issues to a specific database resulting in higher confidence diagnosis and improved corrective actions.
- **Oracle Cluster Health Advisor Cross-Cluster Analysis Support**
In its previous release, Oracle Cluster Health Advisor analyzed each cluster node individually. Oracle Cluster Health Advisor could only detect whether the cause of a performance or availability issue was within itself, or external. With the new cross-cluster analysis support, Oracle Cluster Health Advisor can target external issues to a specific cluster node resulting in higher confidence diagnosis and improved corrective actions.

New Features for Oracle Database Quality of Service (QoS) Management

Oracle Database Quality of Service (QoS) Management includes the following new feature:

- The ability to update an active policy set when adding new database services using `qosctl qos_admin_user -updatepolicyset`. This aids automatic provisioning especially in fleet or cloud environments.

Related Topics

- [QOSCTL Utility Reference](#)
QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.
- [Updating a Policy Set to Include a New Database](#)
If a new database is added to your cluster, you can configure Oracle Database QoS Management to manage or monitor the workload on this database.

1

Introduction to Oracle Autonomous Health Framework

Oracle Autonomous Health Framework is a collection of components that analyzes the diagnostic data collected, and proactively identifies issues before they affect the health of your clusters or your Oracle Real Application Clusters (Oracle RAC) databases.

Most of the Oracle Autonomous Health Framework components are already available in Oracle Database 12c release 1 (12.1). In Oracle Database 12c release 2 (12.2), the output of several components is consolidated in the Grid Infrastructure Management Repository (GIMR) and analyzed in real time to detect problematic patterns on the production clusters.

- [Oracle Autonomous Health Framework Problem and Solution Space](#)
Oracle Autonomous Health Framework assists with monitoring, diagnosing, and preventing availability and performance issues.
- [Components of Oracle Autonomous Health Framework](#)
This section describes the diagnostic components that are part of Oracle Autonomous Health Framework.

1.1 Oracle Autonomous Health Framework Problem and Solution Space

Oracle Autonomous Health Framework assists with monitoring, diagnosing, and preventing availability and performance issues.

System administrators can use most of the components in Oracle Autonomous Health Framework interactively during installation, patching, and upgrading. Database administrators can use Oracle Autonomous Health Framework to diagnose operational runtime issues and mitigate the impact of these issues.

- [Availability Issues](#)
Availability issues are runtime issues that threaten the availability of software stack.
- [Performance Issues](#)
Performance issues are runtime issues that threaten the performance of the system.

1.1.1 Availability Issues

Availability issues are runtime issues that threaten the availability of software stack.

Availability issues can result from either software issues (Oracle Database, Oracle Grid Infrastructure, operating system) or the underlying hardware resources (CPU, Memory, Network, Storage).

The components within Oracle Autonomous Health Framework address the following availability issues:

Examples of Server Availability Issues

Server availability issues can cause a server to be evicted from the cluster and shut down all the database instances that are running on the server.

Examples of such issues are:

- **Issue:** Memory stress caused by a server running out of free physical memory, results in the operating system `Swapper` process to run for extended periods of time moving memory to disk. Swapping prevents time-critical cluster processes from running and eventually causing the node to be evicted.
Solution: Memory Guard detects the memory stress in advance and causes work to be drained to free up memory.
- **Issue:** Network congestion on the private interconnect can cause time-critical internode or storage I/O to have excessive latency or dropped packets. This type of failure typically builds up and can be detected early, and corrected or relieved.
Solution: If a change in the server configuration causes this issue, then Cluster Verification Utility (CVU) detects it if the issue persists for more than an hour. However, Oracle Cluster Health Advisor detects the issue within minutes and presents corrective actions.
- **Issue:** Network failures on the private interconnect caused by a pulled cable or failed network interface card (NIC) can immediately result in evicted nodes.
Solution: Although these types of network failures cannot be detected early, the cause can be narrowed down by using Cluster Health Monitor and Oracle Trace File Analyzer to pinpoint the time of the failure and the network interfaces involved.

Examples of Database Availability Issues

Database availability issues can cause an Oracle database or one of the instances of the database to become unresponsive and thus unavailable to users.

Examples of such issues are:

- **Issue:** Runaway queries or hangs can deny critical database resources such as locks, latches, or CPU to other sessions. Denial of critical database resources results in database or an instance of a database being non-responsive to applications.
Solution: Hang Manager detects and automatically resolves these types of hangs. Also, Oracle Cluster Health Advisor detects, identifies, and notifies the database administrator of such hangs and provides an appropriate corrective action.
- **Issue:** Denial-of-service (DoS) attacks, vulnerabilities, or simply software bugs can cause a database or a database instance to be unresponsive.
Solution: Proactive recommendations of known issues and their resolutions provided by Oracle ORAchk can prevent such occurrences. If these issues are not prevented, then automatic collection of logs by Oracle Trace File Analyzer, in addition to data collected by Cluster Health Monitor, can speed up the correction of these issues.
- **Issue:** Configuration changes can cause database outages that are difficult to troubleshoot. For example, incorrect permissions on the `oracle.bin` file can prevent session processes from being created.

Solution: Use Cluster Verification Utility and Oracle ORAchk to speed up identification and correction of these types of issues. You can generate a diff report using Oracle ORAchk to see a baseline comparison of two reports and a list of differences. You can also view configuration reports created by Cluster Verification Utility to verify whether your system meets the criteria for an Oracle installation.

1.1.2 Performance Issues

Performance issues are runtime issues that threaten the performance of the system.

Performance issues can result from either software issues (bugs, configuration problems, data contention, and so on) or client issues (demand, query types, connection management, and so on).

Server and database performance issues are intertwined and difficult to separate. It is easier to categorize them by their origin: database server or client.

Examples of Database Server Performance Issues

- **Issue:** Deviations from best practices in configuration can cause database server performance issues.
Solution: Oracle ORAchk detects configuration issues when Oracle ORAchk runs periodically and notifies the database administrator of the appropriate corrective settings.
- **Issue:** Bottlenecked resources or poorly constructed SQL statements can cause database server performance issues.
Solution: Oracle Database Quality of Service (QoS) Management flags these issues and generates notifications when the issues put Service Level Agreements (SLAs) at risk. Oracle Cluster Health Advisor detects when the issues exceed normal operating conditions and notifies the database administrator with corrective actions.
- **Issue:** A session can cause other sessions to slow down waiting for the blocking session to release its resource or complete its work.
Solution: Hang Manager detects these chains of sessions and automatically kills the root holder session to relieve the bottleneck.
- **Issue:** Unresolved known issues or unpatched bugs can cause database server performance issues.
Solution: These issues can be detected through the automatic Oracle ORAchk reports and flagged with associated patches or workarounds. Oracle ORAchk is regularly enhanced to include new critical issues, either in existing products or in new product areas.

Examples of Performance Issues Caused by Database Client

- **Issue:** When a server is hosting more database instances than its resources and client load can manage, performance suffers because of waits for CPU, I/O, or memory.
Solution: Oracle ORAchk and Oracle Database QoS Management detect when these issues are the result of misconfiguration such as oversubscribing of CPUs, memory, or background processes. Oracle ORAchk and Oracle Database QoS Management notify you with corrective actions.

- **Issue:** Misconfigured parameters such as SGA and PGA allocation, number of sessions or processes, CPU counts, and so on, can cause database performance degradation.
Solution: Oracle ORAchk and Oracle Cluster Health Advisor detect the settings and consequences respectively and notify you automatically with recommended corrective actions.
- **Issue:** A surge in client connections can exceed the server or database capacity, causing timeout errors and other performance problems.
Solution: Oracle Database QoS Management and Oracle Cluster Health Advisor automatically detect the performance degradation. Also, Oracle Database QoS Management and Oracle Cluster Health Advisor notify you with corrective actions to relieve the bottleneck and restore performance.

1.2 Components of Oracle Autonomous Health Framework

This section describes the diagnostic components that are part of Oracle Autonomous Health Framework.

- [Introduction to Oracle ORAchk and Oracle EXAchk](#)
Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.
- [Introduction to Cluster Health Monitor](#)
Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.
- [Introduction to Oracle Trace File Analyzer](#)
Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.
- [Introduction to Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.
- [Introduction to Memory Guard](#)
Memory Guard is an Oracle Real Application Clusters (Oracle RAC) environment feature to monitor the cluster nodes to prevent node stress caused by the lack of memory.
- [Introduction to Hang Manager](#)
Hang Manager is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves hangs and keeps the resources available.
- [Introduction to Oracle Database Quality of Service \(QoS\) Management](#)
Oracle Database Quality of Service (QoS) Management manages the resources that are shared across applications.

1.2.1 Introduction to Oracle ORAchk and Oracle EXAchk

Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.

Oracle ORAchk and Oracle EXAchk:

- Automates risk identification and proactive notification before your business is impacted
- Runs health checks based on critical and reoccurring problems
- Presents high-level reports about your system health risks and vulnerabilities to known issues
- Enables you to drill-down specific problems and understand their resolutions
- Enables you to schedule recurring health checks at regular intervals
- Sends email notifications and diff reports while running in daemon mode
- Integrates the findings into Oracle Health Check Collections Manager and other tools of your choice
- Runs in your environment with no need to send anything to Oracle

You have access to Oracle ORAchk and Oracle EXAchk as a value add-on to your existing support contract. There is no additional fee or license required to run Oracle ORAchk and Oracle EXAchk.

Use Oracle EXAchk for Oracle Engineered Systems except for Oracle Database Appliance. For all other systems, use Oracle ORAchk.

Run health checks for Oracle products using the command-line options.

Related Topics

- [Oracle ORAchk and Oracle EXAchk Command-Line Options](#)
Most command-line options apply to both Oracle ORAchk and Oracle EXAchk. Use the command options to control the behavior of Oracle ORAchk and Oracle EXAchk.
- [Analyzing Risks and Complying with Best Practices](#)
Use configuration audit tools Oracle ORAchk and Oracle EXAchk to assess your Oracle Engineered Systems and non-Engineered Systems for known configuration problems and best practices.

1.2.2 Introduction to Cluster Health Monitor

Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.

Enabled by default, Cluster Health Monitor:

- Assists node eviction analysis
- Logs all process data locally
- Enables you to define pinned processes
- Listens to CSS and GIPC events
- Categorizes processes by type
- Supports plug-in collectors such as traceroute, netstat, ping, and so on
- Provides CSV output for ease of analysis

Cluster Health Monitor serves as a data feed for other Oracle Autonomous Health Framework components such as Oracle Cluster Health Advisor and Oracle Database Quality of Service Management.

Related Topics

- [Collecting Operating System Resources Metrics](#)
Use Cluster Health Monitor to collect diagnostic data to analyze performance degradation or failures of critical operating system resources.

1.2.3 Introduction to Oracle Trace File Analyzer

Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.

Enabled by default, Oracle Trace File Analyzer:

- Provides comprehensive first failure diagnostics collection
- Efficiently collects, packages, and transfers diagnostic data to Oracle Support
- Reduces round trips between customers and Oracle

Oracle Trace File Analyzer reduces the time required to obtain the correct diagnostic data, which eventually saves your business money.

Related Topics

- [Oracle Trace File Analyzer](#)
Oracle Trace File Analyzer helps you collect and analyze diagnostic data.

1.2.4 Introduction to Oracle Cluster Health Advisor

Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.

Oracle Cluster Health Advisor is integrated into Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager.

Oracle Cluster Health Advisor does the following:

- Detects node and database performance problems
- Provides early-warning alerts and corrective action
- Supports on-site calibration to improve sensitivity

In Oracle Database 12c release 2 (12.2.0.1), Oracle Cluster Health Advisor supports the monitoring of two critical subsystems of Oracle Real Application Clusters (Oracle RAC): the database instance and the host system. Oracle Cluster Health Advisor determines and tracks the health status of the monitored system. It periodically samples a wide variety of key measurements from the monitored system.

Over a hundred database and cluster node problems have been modeled, and the specific operating system and Oracle Database metrics that indicate the development or existence of these problems have been identified. This information is used to construct a trained, calibrated model that is based on a normal operational period of the target system.

Oracle Cluster Health Advisor runs an analysis multiple times a minute. Oracle Cluster Health Advisor estimates an expected value of an observed input based on the default model. Oracle Cluster Health Advisor then performs anomaly detection for each input based on the difference between observed and expected values. If sufficient inputs associated with a specific problem are abnormal, then Oracle Cluster Health Advisor raises a warning and generates an immediate targeted diagnosis and corrective action.

Oracle Cluster Health Advisor models are conservative to prevent false warning notifications. However, the default configuration may not be sensitive enough for critical production systems. Therefore, Oracle Cluster Health Advisor provides an onsite model calibration capability to use actual production workload data to form the basis of its default setting and increase the accuracy and sensitivity of node and database models.

Oracle Cluster Health Advisor stores the analysis results, along with diagnosis information, corrective action, and metric evidence for later triage, in the Grid Infrastructure Management Repository (GIMR). Oracle Cluster Health Advisor also sends warning messages to Enterprise Manager Cloud Control using the Oracle Clusterware event notification protocol.

You can also use Oracle Cluster Health Advisor to diagnose and triage past problems. You specify the past dates through Oracle Enterprise Manager Cloud Control (EMCC) Incident Manager or through the command-line interface CHACTL. Manage the capability of Oracle Cluster Health Advisor to review past problems by configuring the retention setting for Oracle Cluster Health Advisor's tablespace in the Grid Infrastructure Management Repository (GIMR). The default retention period is 72 hours.

Related Topics

- [Oracle Trace File Analyzer](#)
Oracle Trace File Analyzer helps you collect and analyze diagnostic data.

1.2.5 Introduction to Memory Guard

Memory Guard is an Oracle Real Application Clusters (Oracle RAC) environment feature to monitor the cluster nodes to prevent node stress caused by the lack of memory.

Enabled by default, Memory Guard:

- Analyzes over-committed memory conditions once in every minute
- Issues alert if any server is at risk
- Protects applications by automatically closing the server to new connections
- Stops all CRS-managed services transactionally on the server
- Re-opens server to connections once the memory pressure has subsided

Enterprise database servers can use all available memory due to too many open sessions or runaway workloads. Running out of memory can result in failed transactions or, in extreme cases, a restart of the node and the loss of availability of resources for your applications.

Memory Guard autonomously collects metrics on memory of every node from Cluster Health Monitor to determine if the nodes have insufficient memory. If the memory is insufficient, then Memory Guard prevents new database sessions from being created

allowing the existing workload to complete and free their memory. New sessions are started automatically when the memory stress is relieved.

Related Topics

- [Resolving Memory Stress](#)
Memory Guard continuously monitors and ensures the availability of cluster nodes by preventing the nodes from being evicted when the nodes are stressed due to lack of memory.

1.2.6 Introduction to Hang Manager

Hang Manager is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves hangs and keeps the resources available.

Enabled by default, Hang Manager:

- Reliably detects database hangs and deadlocks
- Autonomously resolves database hangs and deadlocks
- Supports Oracle Database QoS Performance Classes, Ranks, and Policies to maintain SLAs
- Logs all detections and resolutions
- Provides SQL interface to configure sensitivity (Normal/High) and trace file sizes

A database hangs when a session blocks a chain of one or more sessions. The blocking session holds a resource such as a lock or latch that prevents the blocked sessions from progressing. The chain of sessions has a root or a final blocker session, which blocks all the other sessions in the chain. Hang Manager resolves these issues autonomously by detecting and resolving the hangs.

Related Topics

- [Resolving Database and Database Instance Hangs](#)
Hang Manager preserves the database performance by resolving hangs and keeping the resources available.

1.2.7 Introduction to Oracle Database Quality of Service (QoS) Management

Oracle Database Quality of Service (QoS) Management manages the resources that are shared across applications.

Oracle Database Quality of Service (QoS) Management:

- Requires 12.1.0.2+ Oracle Grid Infrastructure
- Delivers Key Performance Indicators cluster-wide dashboard
- Phase in with Measure, Monitor, then Management Modes

Oracle Database Quality of Service (QoS) Management adjusts the system configuration to keep the applications running at the performance levels needed by your business.

Many companies are consolidating and standardizing their data center computer systems. Instead of using individual servers for each application, the companies run multiple applications on clustered databases. In addition, migration of applications to

the Internet has introduced the problem of managing an **open workload**. An open workload is subject to demand surges that can overload a system. Over loading a system results in a new type of application failure that cannot be fully anticipated or planned for. To keep the applications available and performing within their target service levels in this type of environment, you must:

- Pool resources
- Have management tools that detect performance bottlenecks in real time
- Reallocate resources to meet the change in demand

Oracle Database QoS Management responds gracefully to changes in system configuration and demand, thus avoiding more oscillations in the performance levels of your applications.

Oracle Database QoS Management monitors the performance of each **work request** on a target system. Oracle Database QoS Management starts to track a work request from the time a work request tries to establish a connection to the database using a database service. The time required to complete a work request or the response time is the time from when the request for data was initiated and when the data request is completed. The response time is also known as the **end-to-end response time**, or **round-trip** time. By accurately measuring the two components of response time, Oracle Database QoS Management quickly detects bottlenecks in the system. Oracle Database QoS Management then suggests reallocating resources to relieve a **bottleneck**, thus preserving or restoring service levels.

Oracle Database QoS Management manages the resources on your system so that:

- When sufficient resources are available to meet the demand, business-level performance requirements for your applications are met, even if the workload changes
- When sufficient resources are *not* available to meet the demand, Oracle Database QoS Management attempts to satisfy the more critical business performance requirements at the expense of less critical performance requirements

Related Topics

- [Monitoring and Managing Database Workload Performance](#)
Oracle Database Quality of Service (QoS) Management is an automated, policy-based product that monitors the workload requests for an entire system.

Part I

Analyzing the Cluster Configuration

You can use tools in the Autonomous Health Framework to analyze your cluster configuration.

- [Analyzing Risks and Complying with Best Practices](#)
Use configuration audit tools Oracle ORAchk and Oracle EXAchk to assess your Oracle Engineered Systems and non-Engineered Systems for known configuration problems and best practices.
- [Proactively Detecting and Diagnosing Performance Issues for Oracle RAC](#)
Oracle Cluster Health Advisor provides system and database administrators with early warning of pending performance issues, and root causes and corrective actions for Oracle RAC databases and cluster nodes.

2

Analyzing Risks and Complying with Best Practices

Use configuration audit tools Oracle ORAchk and Oracle EXAchk to assess your Oracle Engineered Systems and non-Engineered Systems for known configuration problems and best practices.

- [Using Oracle ORAchk and Oracle EXAchk to Automatically Check for Risks and System Health](#)
Oracle recommends that you use the daemon process to schedule recurring health checks at regular intervals.
- [Email Notification and Health Check Report Overview](#)
The following sections provide a brief overview about email notifications and sections of the HTML report output.
- [Configuring Oracle ORAchk and Oracle EXAchk](#)
To configure Oracle ORAchk and Oracle EXAchk, use the procedures explained in this section.
- [Using Oracle ORAchk and Oracle EXAchk to Manually Generate Health Check Reports](#)
This section explains the procedures to manually generate health check reports.
- [Managing the Oracle ORAchk and Oracle EXAchk Daemons](#)
This section explains the procedures to manage Oracle ORAchk and Oracle EXAchk daemons.
- [Tracking Support Incidents](#)
The **Incidents** tab gives you a complete system for tracking support incidents.
- [Tracking File Attribute Changes and Comparing Snapshots](#)
Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.
- [Collecting and Consuming Health Check Data](#)
Oracle Health Check Collections Manager for Oracle Application Express 4.2 provides you an enterprise-wide view of your health check collection data.
- [Locking and Unlocking Storage Server Cells](#)
Beginning with version 12.1.0.2.7, use Oracle EXAchk to lock and unlock storage server cells.
- [Integrating Health Check Results with Other Tools](#)
Integrate health check results from Oracle ORAchk and Oracle EXAchk into Enterprise Manager and other third-party tools.
- [Troubleshooting Oracle ORAchk and Oracle EXAchk](#)
To troubleshoot and fix Oracle ORAchk and Oracle EXAchk issues, follow the steps explained in this section.

Related Topics

- [Introduction to Oracle ORAchk and Oracle EXAchk](#)
Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.

2.1 Using Oracle ORAchk and Oracle EXAchk to Automatically Check for Risks and System Health

Oracle recommends that you use the daemon process to schedule recurring health checks at regular intervals.



Note:

Daemon mode is supported only on the Linux and Solaris operating systems.

Configure the daemon to:

- Schedule recurring health checks at regular interval
- Send email notifications when the health check runs complete, clearly showing any differences since the last run
- Purge collection results after a pre-determined period
- Check and send email notification about stale passwords
- Store multiple profiles for automated health check runs
- Restart automatically if the *server* or *node* where it is running restarts



Note:

While running, the daemon answers all the prompts required by subsequent on-demand health checks.

To run on-demand health checks, do not use the daemon process started by others. Run on-demand health checks within the same directory where you have started the daemon.

If you change the system configuration such as adding or removing *servers* or *nodes*, then restart the daemon.

Related Topics

- [Setting and Getting Options for the Daemon](#)
Set the daemon options before you start the daemon. Reset the daemon options anytime after starting the daemon.
- [Starting and Stopping the Daemon](#)
Start and stop the daemon and force the daemon to stop a health check run.

- [Querying the Status and Next Planned Daemon Run](#)
Query the status and next automatic run schedule of the running daemon.
- [Configuring the Daemon for Automatic Restart](#)
By default, you must manually restart the daemon if you restart the *server* or *node* on which the daemon is running.

2.2 Email Notification and Health Check Report Overview

The following sections provide a brief overview about email notifications and sections of the HTML report output.

- [First Email Notification](#)
After completing health check runs, the daemon emails the assessment report as an HTML attachment to all users that you have specified in the `NOTIFICATION_EMAIL` list.
- [What does the Health Check Report Contain?](#)
Health check reports contain the health status of each system grouped under different sections of the report.
- [Subsequent Email Notifications](#)
For the subsequent health check runs after the first email notification, the daemon emails the summary of differences between the most recent runs.

Related Topics

- [Generating a Diff Report](#)
The diff report attached to the previous email notification shows a summary of differences between the most recent runs.

2.2.1 First Email Notification

After completing health check runs, the daemon emails the assessment report as an HTML attachment to all users that you have specified in the `NOTIFICATION_EMAIL` list.

2.2.2 What does the Health Check Report Contain?

Health check reports contain the health status of each system grouped under different sections of the report.

The HTML report output contains the following:

- Health score
- Summary of health check runs
- Table of contents
- Controls for report features
- Findings
- Recommendations

Details of the report output are different on each system. The report is dynamic, and therefore the tools display certain sections only if applicable.

System Health Score and Summary

System Health Score and Summary report provide:

- A high-level health score based on the number of passed or failed checks
- A summary of health check run includes:
 - Name, for example, Cluster Name
 - Version of the operating system kernel
 - Path, version, name of homes, for example, CRS, DB, and EM Agent
 - Version of the component checked, for example, Exadata
 - Number of nodes checked, for example, database server, storage servers, InfiniBand switches
 - Version of Oracle ORAchk and Oracle EXAchk
 - Name of the collection output
 - Date and time of collection
 - Duration of the check
 - Name of the user who ran the check, for example, `root`
 - How long the check is valid

Table of Contents and Report Feature

The **Table of Contents** section provides links to major sections in the report:

- Database Server
- Storage Server
- InfiniBand Switch
- Cluster Wide
- Maximum Availability Architecture (MAA) Scorecard
- Infrastructure Software and Configuration Summary
- Findings needing further review
- Platinum Certification
- System-wide Automatic Service Request (ASR) health check
- Skipped Checks
- Top 10 Time Consuming Checks

The **Report Feature** section enables you to:

- Filter checks based on their statuses
- Select the regions
- Expand or collapse all checks
- View check IDs
- Remove findings from the report
- Get a printable view

Report Findings

The **Report Findings** section displays the result of each health check grouped by technology components, such as Database Server, Storage Server, InfiniBand Switch, and Cluster Wide.

Each section shows:

- Check status (FAIL, WARNING, INFO, or PASS)
- Type of check
- Check message
- Where the check was run
- Link to expand details for further findings and recommendation

Click **View** for more information about the health check results and the recommendations.

- What to do to solve the problem
- Where the recommendation applies
- Where the problem does not apply
- Links to relevant documentation or My Oracle Support notes
- Example of data on which the recommendation is based

Maximum Availability Architecture (MAA) Score Card

Maximum Availability Architecture (MAA) Score Card displays the recommendations for the software installed on your system.

The details include:

- Outage Type
- Status of the check
- Description of the problem
- Components found
- Host location
- Version of the components compared to the recommended version
- Status based on comparing the version found to the recommended version

2.2.3 Subsequent Email Notifications

For the subsequent health check runs after the first email notification, the daemon emails the summary of differences between the most recent runs.

Specify a list of comma-delimited email addresses in the `NOTIFICATION_EMAIL` option.

The email notification contains:

- System Health Score of this run compared to the previous run
- Summary of number of checks that were run and the differences between runs

- Most recent report result as attachment
- Previous report result as attachment
- Diff report as attachment

2.3 Configuring Oracle ORAchk and Oracle EXAchk

To configure Oracle ORAchk and Oracle EXAchk, use the procedures explained in this section.

- [Deciding Which User Should Run Oracle ORAchk or Oracle EXAchk](#)
Run health checks as `root`. Also, run health checks as the Oracle Database home owner or the Oracle Grid Infrastructure home owner.
- [Handling of Root Passwords](#)
Handling of `root` passwords depends on whether you have installed the Expect utility.
- [Configuring Email Notification System](#)
Oracle Health Check Collections Manager provides an email notification system that users can subscribe to.

2.3.1 Deciding Which User Should Run Oracle ORAchk or Oracle EXAchk

Run health checks as `root`. Also, run health checks as the Oracle Database home owner or the Oracle Grid Infrastructure home owner.

Most health checks do not require `root` access. However, you need `root` privileges to run a subset of health checks.

To run `root` privilege checks, Oracle ORAchk uses the script `root_orachk.sh` and Oracle EXAchk uses the script `root_exachk.sh`.

By default, the `root_orachk.sh` and `root_exachk.sh` scripts are created in the `$HOME` directory used by Oracle ORAchk and Oracle EXAchk. Change the directory by setting the environment variable `RAT_ROOT_SH_DIR`.

Specify a location for `sudo` remote access as follows:

```
export RAT_ROOT_SH_DIR=/mylocation
```

Add an entry in the `/etc/sudoers` as follows:

```
oracle ALL=(root) NOPASSWD:/mylocation/root_orachk.sh
```

For security reasons, create the `root` scripts outside of the standard temporary directory in a custom directory.

To decide which user to run Oracle ORAchk and Oracle EXAchk:

1. Specify the custom directory using the `RAT_ROOT_SH_DIR` environment variable.

```
export RAT_ROOT_SH_DIR=/orahome/oradb/
```

2. Specify a location for `sudo` remote access.

```
export RAT_ROOT_SH_DIR=/mylocation
```

3. Add an entry in the `/etc/sudoers` file.

```
oracle ALL=(root) NOPASSWD:/mylocation/root_orachk.sh
```

 **Note:**

Specify full paths for the entries in the `/etc/sudoers` file. Do not use environment variables.

4. (recommended) Run Oracle ORAchk and Oracle EXAchk as `root`.

Use `root` user credentials to run Oracle ORAchk and Oracle EXAchk.

The Oracle ORAchk and Oracle EXAchk processes that run as `root`, perform user lookups for the users who own the Oracle Database home and Oracle Grid Infrastructure home. If `root` access is not required, then the Oracle ORAchk and Oracle EXAchk processes use the `su` command to run health checks as the applicable Oracle Database home user or Oracle Grid Infrastructure home user. Accounts with lower privileges cannot have elevated access to run health checks that require `root` access.

Running health checks as `root` has advantages in role-separated environments or environments with more restrictive security.

5. Run Oracle ORAchk and Oracle EXAchk as Oracle Database home owner or Oracle Grid Infrastructure home owner:

Use Oracle Database home owner or Oracle Grid Infrastructure home owner credentials to run Oracle ORAchk and Oracle EXAchk.

The user who runs Oracle ORAchk and Oracle EXAchk must have elevated access as `root` to run health checks that need `root` access.

Running health checks as Oracle Database home owner or Oracle Grid Infrastructure home owner requires multiple runs in role-separated environments. More restrictive security requirements do not permit elevated access.

There are several other options:

- Skip the checks that require `root` access.
- Specify the `root` user ID and password when prompted.
- Configure `sudo`.

If you are using `sudo`, then add an entry for the `root` script, located in `$HOME` in the `/etc/sudoers` file that corresponds to the user who is running the health checks.

To determine what `$HOME` is set to, run the `echo $HOME` command.

For example:

```
user ALL=(root) NOPASSWD:/root/root_orachk.sh
```

```
user ALL=(root) NOPASSWD:/root/root_exachk.sh
```

- Pre-configure passwordless SSH connectivity.

2.3.2 Handling of Root Passwords

Handling of `root` passwords depends on whether you have installed the Expect utility.

Expect automates interactive applications such as Telnet, FTP, `passwd`, `fsck`, `rlogin`, `tip`, and so on.

To handle root passwords:

1. If you have installed the Expect utility, then specify the `root` password when you run the health checks for the first time.

The Expect utility stores the password and uses the stored password for subsequent sessions.

The Expect utility prompts you to check if the `root` password is same for all the remote components such as databases, switches, and so on.

2. Specify the password only once if you have configured the same `root` password for all the components.

If `root` password is not same for all the components, then the Expect utility prompts you to validate `root` password every time you run the health checks.

If you enter the password incorrectly or the password is changed between the time it is entered and used, then Oracle ORAchk and Oracle EXAchk:

- Notify you
- Skip relevant checks

3. Run the health checks after resolving the issues.

If Oracle ORAchk and Oracle EXAchk skip any of the health checks, then the tools log details about the skipped checks in the report output.

Related Topics

- [Expect - Expect - Home Page](#)

2.3.3 Configuring Email Notification System

Oracle Health Check Collections Manager provides an email notification system that users can subscribe to.

The setup involves:

- Configuring the email server, port, and the frequency of email notifications.
- Registering the email address



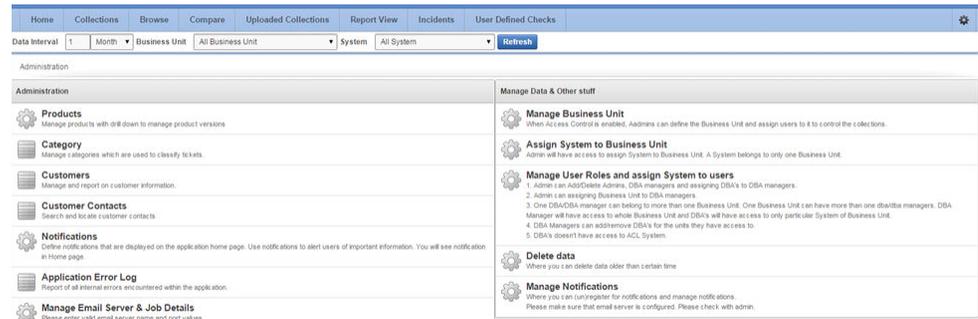
Note:

Only the users who are assigned Admin role can manage **Email Notification Server and Job details**.

To configure the email notification system:

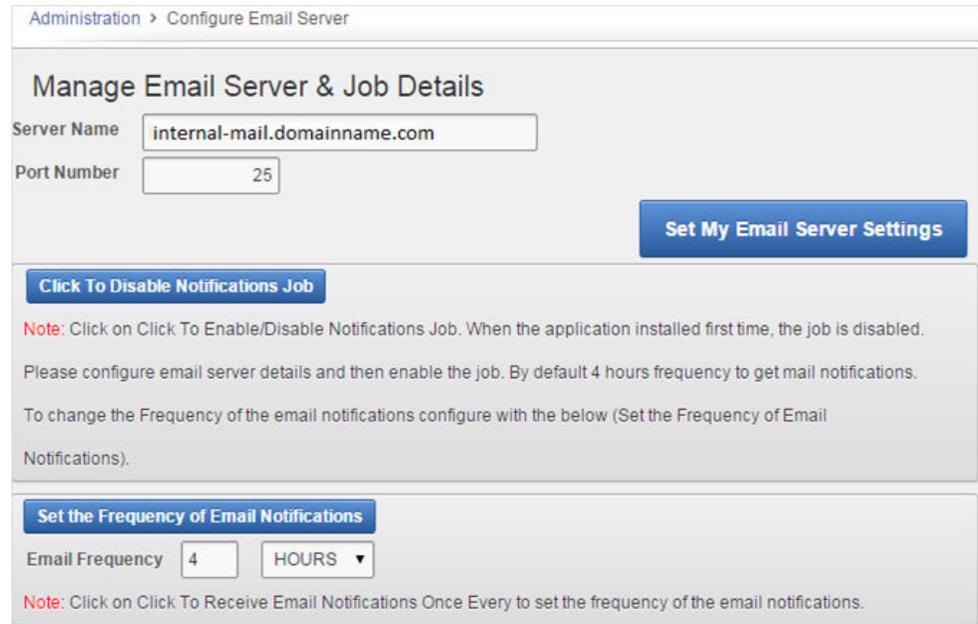
1. Log in to Oracle Health Check Collections Manager, and then click **Administration** at the upper-right corner.

Figure 2-1 Oracle Health Check Collections Manager - Administration



2. Under **Administration**, click **Manage Email Server & Job Details**.

Figure 2-2 Oracle Health Check Collections Manager - Configure Email Server



- a. Specify a valid **Email Server Name**, **Port Number**, and then click **Set My Email Server Settings**.
- b. Set **Email Notification Frequency** as per your needs.
See the **Notification Job Run Details** on the same page.

Figure 2-3 Oracle Health Check Collections Manager - Notification Job Run status details

Notifications Job Run Details		
Run On	Status	Error Code
31-MAY-14 04.00.09.397435 AM -07:00	SUCCEEDED	0
31-MAY-14 12.00.00.699009 PM -07:00	SUCCEEDED	0
03-JUN-14 04.00.02.741378 AM -07:00	SUCCEEDED	0
01-JUN-14 12.00.00.640109 PM -07:00	SUCCEEDED	0
31-MAY-14 08.00.02.529193 AM -07:00	SUCCEEDED	0

- Go back to the **Administration** page, and click **Manage Notifications**.

Figure 2-4 Oracle Health Check Collections Manager - Manage Notifications

Administration > Manage Notifications

Register For Email Notifications

Email Id Subscribe/Unsubscribe My Mail Notifications

Collection Notifications

New Collections Without Comparisons

Collections Regressed with Warnings

Collections that Improved with Passes

Collections Regressed with Failures

ORAchk CM Tablespace Notifications

ORAchk CM space in data base falls below 100MB

Note: Please make sure the email is valid. If ACL system is enabled, All subscribed users will receive notifications for the systems that they have access on their notification preferences.

Test your email settings

Use the Test email button to verify proper email delivery. If there is a problem please contact your administrator.

- If you are configuring for the first time, then enter your email address. Subsequent access to **Manage Notifications** page shows your email address automatically.
- By default, **Subscribe/Unsubscribe My Mail Notifications** is checked. Leave as is.
- Under **Collection Notifications**, choose the type of collections for which you want to receive notifications.
- Select to receive notification when the available space in ORAchk CM Tablespace falls below 100 MB.
- Validate the notification delivery by clicking **Test** under **Test your email settings**.
If the configuration is correct, then you must receive an email. If you do not receive an email, then check with your administrator.

Following is the sample notification:

From: username@example.com
 Sent: Thursday, January 28, 2016 12:21 PM
 To: username@example.com
 Subject: Test Mail From Collection Manager

Testing Collection Manager Email Notification System

f. Click **Submit**.

 **Note:**

Manage Notifications section under the **Administration** menu is available for all users irrespective of the role.

If the ACL system is enabled, then the registered users receive notifications for the systems that they have access to. If the ACL system is not configured, then all the registered users receive all notifications.

Depending on the selections, you made under **Collection Notifications** section, you receive an email with Subject: Collection Manager Notifications containing application URL with results.

Figure 2-5 Oracle Health Check Collections Manager - Sample Email Notification

From: username@domainname.com [mailto:username@domainname.com]
 Sent: Wednesday, February 03, 2016 1:24 AM
 To: username@domainname.com
 Subject: Collection Manager Notifications

Found Diff for the following collections

BU Name	System Name	Previous Collection	Current Collection	Collection DifferenceType	Comments
DEFAULT	cloud00290036	orachk_cloud0029_SOLTEN_010416_060310	orachk_cloud0029_SOLTEN_010416_072847	Collections Regressed with Failures	Click here for details
DEFAULT	cloud00290036	orachk_cloud0029_SOLTEN_123015_074624	orachk_cloud0029_SOLTEN_010416_060310	Collections Regressed with Failures	Click here for details
DEFAULT	cloud00290036	orachk_cloud0029_SOLTEN_123015_062009	orachk_cloud0029_SOLTEN_123015_074624	Collections Regressed with Warnings	Click here for details
DEFAULT	cloud00290036	orachk_cloud0029_SOLTEN_010416_072847	orachk_cloud0029_SOLTEN_010416_125702	Collections Regressed with Warnings	Click here for details

Under **Comments** column, click the **Click here** links for details. Click the respective URLs, authenticate, and then view respective comparison report.

Figure 2-6 Oracle Health Check Collections Manager - Sample Diff Report

The screenshot displays the Oracle Health Check Collections Manager interface. At the top, there are navigation tabs: Home, Collections, Compare, Uploaded Collections, Report View, Incidents, and User Defined Checks. Below these are filters for Data Interval (1 Hour), Business Unit (All Business Unit), System (All System), and DB Version. A 'Refresh' button is present. The main content area is titled 'Health Checks Baseline Comparison Report' and shows two collection details side-by-side. Below this is a table of 'Checks Matched' with the following data:

Check Name	Status	StatusMsg	Host Name	DB Name	Instance Name
ASIM Important INFO	INFO	Important Automatic Storage Management (ASM) Notes and Technical White Papers	rws1270029		NA
ASH flush status	PASS	AWR_FLUSH_EMERGENCY_COUNT value is equal to ZERO for SOLTEN	rws1270029	SOLTEN	NA
DBRM Check Status	WARNING	DBRM is not configured for SOLTEN	rws1270029	SOLTEN	NA
Ensure db_unique_name is unique across the enterprise [primary]	FAIL	DB_UNIQUE_NAME on primary has not been modified from the default, confirm that database name is unique across your Oracle enterprise for SOLTEN	rws1270029	SOLTEN	NA
ROBMS software owner across cluster	PASS	ROBMS software owner matches across cluster	rws1270031	NA	NA

2.4 Using Oracle ORAchk and Oracle EXAchk to Manually Generate Health Check Reports

This section explains the procedures to manually generate health check reports.

- [Running Health Checks On-Demand](#)
Usually, health checks run at scheduled intervals. However, Oracle recommends that you run health checks on-demand when needed.
- [Running Health Checks in Silent Mode](#)
Run health checks automatically by scheduling them with the Automated Daemon Mode operation.
- [Running On-Demand With or Without the Daemon](#)
When running on-demand, if the daemon is running, then the daemon answers all prompts where possible including the passwords.
- [Generating a Diff Report](#)
The diff report attached to the previous email notification shows a summary of differences between the most recent runs.
- [Sending Results by Email](#)
Optionally email the HTML report to one or more recipients using the `-sendemail` option.

2.4.1 Running Health Checks On-Demand

Usually, health checks run at scheduled intervals. However, Oracle recommends that you run health checks on-demand when needed.

Examples of when you must run health checks on-demand:

- Pre- or post-upgrades
- Machine relocations from one subnet to another

- Hardware failure or repair
- Problem troubleshooting
- In addition to go-live testing

To start on-demand health check runs, log in to the system as an appropriate user, and then run an appropriate tool. Specify the options to direct the type of run that you want.

```
$ ./orachk
```

```
$ ./exachk
```

 **Note:**

To avoid problems while running the tool from terminal sessions on a network attached workstation or laptop, consider running the tool using VNC. If there is a network interruption, then the tool continues to process to completion. If the tool fails to run, then re-run the tool. The tool does not resume from the point of failure.

Output varies depending on your environment and options used:

- The tool starts discovering your environment
- If you have configured passwordless SSH equivalency, then the tool does not prompt you for passwords
- If you have not configured passwordless SSH for a particular component at the required access level, then the tool prompts you for password
- If the daemon is running, then the commands are sent to the daemon process that answers all prompts, such as selecting the database and providing passwords
- If the daemon is not running, then the tool prompts you for required information, such as which database you want to run against, the required passwords, and so on
- The tool investigates the status of the discovered components

 **Note:**

If you are prompted for passwords, then the Expect utility runs when available. In this way, the passwords are gathered at the beginning, and the Expect utility supplies the passwords when needed at the root password prompts. The Expect utility being supplying the passwords enables the tool to continue without the need for further input. If you do not use the Expect utility, then closely monitor the run and enter the passwords interactively as prompted.

Without the Expect utility installed, you must enter passwords many times depending on the size of your environment. Therefore, Oracle recommends that you use the Expect utility.

While running pre- or post-upgrade checks, Oracle ORAchK and Oracle EXAchK automatically detect databases that are registered with Oracle Clusterware and presents the list of databases to check.

Run the pre-upgrade checks during the upgrade planning phase. Oracle ORAchK and Oracle EXAchK prompt you for the version to which you are planning to upgrade:

```
$ ./orachk -u -o pre
```

```
$ ./exachk -u -o pre
```

After upgrading, run the post-upgrade checks:

```
$ ./orachk -u -o post
```

```
$ ./exachk -u -o post
```

- The tool starts collecting information across all the relevant components, including the remote nodes.
- The tool runs the health checks against the collected data and displays the results.
- After completing the health check run, the tool points to the location of the detailed HTML report and the `.zip` file that contains more output.

Related Topics

- [Running On-Demand With or Without the Daemon](#)
When running on-demand, if the daemon is running, then the daemon answers all prompts where possible including the passwords.
- [Sending Results by Email](#)
Optionally email the HTML report to one or more recipients using the `-sendemail` option.
- [Expect - Expect - Home Page](#)

2.4.2 Running Health Checks in Silent Mode

Run health checks automatically by scheduling them with the Automated Daemon Mode operation.

Note:

Silent mode operation is maintained for backwards compatibility for the customers who were using it before the daemon mode was available. Silent mode is limited in the checks it runs and Oracle does not actively enhance it any further.

- Running health checks in silent mode using the `-s` option does not run any checks on the storage servers and switches.
- Running health checks in silent mode using the `-s` option excludes checks on database server that require `root` access. Also, does not run any checks on the storage servers and database servers.

To run health checks silently, configure passwordless SSH equivalency. It is not required to run remote checks, such as running against a single-instance database.

When health checks are run silently, output is similar to that described in On-Demand Mode Operation.

 **Note:**

If not configured to run in silent mode operation on an Oracle Engineered System, then the tool does not perform storage server or InfiniBand switch checks.

Including Health Checks that Require root Access

Run as `root` or configure `sudo` access to run health checks in silent mode and include checks that require `root` access.

To run health checks including checks that require `root` access, use the `-s` option followed by other required options:

```
$ ./orachk -s
```

```
$ ./exachk -s
```

Excluding Health Checks that Require root Access

To run health checks excluding checks that require `root` access, use the `-S` option followed by other required options:

```
$ ./orachk -S
```

```
$ ./exachk -S
```

Related Topics

- [Using Oracle ORAchk and Oracle EXAchk to Automatically Check for Risks and System Health](#)
Oracle recommends that you use the daemon process to schedule recurring health checks at regular intervals.
- [Running Health Checks On-Demand](#)
Usually, health checks run at scheduled intervals. However, Oracle recommends that you run health checks on-demand when needed.

2.4.3 Running On-Demand With or Without the Daemon

When running on-demand, if the daemon is running, then the daemon answers all prompts where possible including the passwords.

To run health checks on-demand with or without the daemon:

1. To run health checks on-demand if the daemon is running, then use:

```
$ ./orachk
```

```
$ ./exachk
```

2. To avoid connecting to the daemon process, meaning the tool to interactively prompt you as required, use the `-nodaemon` option.

```
$ ./orachk -nodaemon
```

```
$ ./exachk -nodaemon
```

 **Note:**

Daemon mode is supported only on the Linux and Solaris operating systems.

 **Note:**

If you are running database pre-upgrade checks (`-u -o pre`) and if the daemon is running, then you must use the `-nodaemon` option.

2.4.4 Generating a Diff Report

The diff report attached to the previous email notification shows a summary of differences between the most recent runs.

To identify the changes since the last run:

1. Run the following command:

```
$ ./orachk -diff report_1 report_2
```

Review the diff report to see a baseline comparison of the two reports and then a list of differences.

2.4.5 Sending Results by Email

Optionally email the HTML report to one or more recipients using the `-sendemail` option.

To send health check run results by email:

1. Specify the recipients in the `NOTIFICATION_EMAIL` environment variable.

```
$ ./orachk -sendemail "NOTIFICATION_EMAIL=email_recipients"
```

```
$ ./exachk -sendemail "NOTIFICATION_EMAIL=email_recipients"
```

Where *email_recipients* is a comma-delimited list of email addresses.

2. Verify the email configuration settings using the `-testemail` option.

2.5 Managing the Oracle ORAchk and Oracle EXAchk Daemons

This section explains the procedures to manage Oracle ORAchk and Oracle EXAchk daemons.

- [Starting and Stopping the Daemon](#)
Start and stop the daemon and force the daemon to stop a health check run.
- [Configuring the Daemon for Automatic Restart](#)
By default, you must manually restart the daemon if you restart the *server* or *node* on which the daemon is running.
- [Setting and Getting Options for the Daemon](#)
Set the daemon options before you start the daemon. Reset the daemon options anytime after starting the daemon.
- [Querying the Status and Next Planned Daemon Run](#)
Query the status and next automatic run schedule of the running daemon.

2.5.1 Starting and Stopping the Daemon

Start and stop the daemon and force the daemon to stop a health check run.

To start and stop the daemon:

1. To start the daemon, use the `-d start` option as follows:

```
$ ./orachk -d start  
$ ./exachk -d start
```

The tools prompt you to provide required information during startup.

2. To stop the daemon, use the `-d stop` option as follows:

```
$ ./orachk -d stop  
$ ./exachk -d stop
```

If a health check run is progress when you run the stop command, then the daemon indicates so and continues running.

3. To force the daemon to stop a health check run, use the `-d stop_client` option:

```
$ ./orachk -d stop_client  
$ ./exachk -d stop_client
```

The daemon stops the health check run and then confirms when it is done. If necessary, stop the daemon using the `-d stop` option.

2.5.2 Configuring the Daemon for Automatic Restart

By default, you must manually restart the daemon if you restart the *server* or *node* on which the daemon is running.

However, if you use the automatic restart option, the daemon restarts automatically after the *server* or *node* reboot.

Only *root* can configure automatic restart.

To configure the daemon to start automatically:

1. To set up daemon automatic restart, use `-initsetup`:

```
$ ./orachk -initsetup
```

```
$ ./exachk -initsetup
```

The tool prompts you to provide the required information during startup.

 **Note:**

Stop the daemon before running `-initsetup`, if the daemon is already running.

2. To query automatic restart status of the daemon, use `-initcheck`:

```
$ ./orachk -initcheck
```

```
$ ./exachk -initcheck
```

3. To remove automatic restart configuration, use `-initrmsetup`:

```
$ ./orachk -initrmsetup
```

```
$ ./exachk -initrmsetup
```

2.5.3 Setting and Getting Options for the Daemon

Set the daemon options before you start the daemon. Reset the daemon options anytime after starting the daemon.

To set the daemon options:

1. Set the daemon options using the `-set` option.

Set an option as follows:

```
$ ./orachk -set "option_1=option_1_value"
```

```
$ ./exachk -set "option_1=option_1_value"
```

Set multiple options using the `name=value` format separated by semicolons as follows:

```
$ ./orachk -set  
"option_1=option_1_value;option_2=option_2_value;option_n=option_n_value"
```

```
$ ./exachk -set
"option_1=option_1_value;option_2=option_2_value;option_n=option_n_value"
```

- **AUTORUN_SCHEDULE**
Schedule recurring health check runs using the `AUTORUN_SCHEDULE` daemon option.
- **AUTORUN_FLAGS**
The `AUTORUN_FLAGS` daemon option determines how health checks are run.
- **NOTIFICATION_EMAIL**
Set the `NOTIFICATION_EMAIL` daemon option to send email notifications to the recipients you specify.
- **collection_retention**
Set the `collection_retention` daemon option to purge health check collection results that are older than a specified number of days.
- **PASSWORD_CHECK_INTERVAL**
The `PASSWORD_CHECK_INTERVAL` daemon option defines the frequency, in hours, for the daemon to validate the passwords entered when the daemon was started the first time.
- **Setting Multiple Option Profiles for the Daemon**
Use only one daemon process for each server. Do not start a single daemon on multiple databases in a cluster, or multiple daemons on the same database.
- **Getting Existing Options for the Daemon**
Query the values that you set for the daemon options.

2.5.3.1 AUTORUN_SCHEDULE

Schedule recurring health check runs using the `AUTORUN_SCHEDULE` daemon option.

To schedule recurring health check runs:

1. Set the `AUTORUN_SCHEDULE` option, as follows:

```
AUTORUN_SCHEDULE=hour minute day month day_of_week
```

where:

- *minute* is 0-59 (Optional. If omitted, then 0 is used)
- *hour* is 0-23
- *day* is 1-31
- *month* is 1-12
- *day_of_week* is 0-6, where 0=Sunday and 6=Saturday

Use the asterisk (*) as a wildcard to specify multiple values separated by commas.

Table 2-1 `AUTORUN_SCHEDULE`

Example	Result
"AUTORUN_SCHEDULE=0,1 5,30,45 * * * *"	Runs every 15 minutes.
"AUTORUN_SCHEDULE=* * * *"	Runs every hour.

Table 2-1 (Cont.) AUTORUN_SCHEDULE

Example	Result
"AUTORUN_SCHEDULE=3 * * 0"	Runs at 3 AM every Sunday.
"AUTORUN_SCHEDULE=2 * 1, 3, 5"	Runs at 2 AM on Monday, Wednesday, and Friday.
"AUTORUN_SCHEDULE=4 1 * *"	Runs at 4 AM on the first day of every month.
"AUTORUN_SCHEDULE=8,2 0 * * 1, 2, 3, 4, 5"	Runs at 8 AM and 8 PM every Monday, Tuesday, Wednesday, Thursday, and Friday.

For example:

```
$ ./orachk -set "AUTORUN_SCHEDULE=3 * * 0"
$ ./exachk -set "AUTORUN_SCHEDULE=3 * * 0"
```

2.5.3.2 AUTORUN_FLAGS

The `AUTORUN_FLAGS` daemon option determines how health checks are run.

To configure how health checks should run:

1. Set the `AUTORUN_FLAGS` option as follows:

```
AUTORUN_FLAGS=flags
```

where, *flags* can be any combination of valid command-line flags.

Table 2-2 AUTORUN_FLAGS

Example	Result
"AUTORUN_FLAGS=-profile dba"	Runs only the dba profile checks.
"AUTORUN_FLAGS=-profile sysadmin -tag sysadmin"	Runs only the dba profile checks and tags the output with the value sysadmin.
-excludeprofile ebs	Runs all checks except the checks in the ebs profile.

For example:

```
$ ./orachk -set "AUTORUN_FLAGS=-profile sysadmin -tag sysadmin"
$ ./exachk -set "AUTORUN_FLAGS=-profile sysadmin -tag sysadmin"
```

2.5.3.3 NOTIFICATION_EMAIL

Set the `NOTIFICATION_EMAIL` daemon option to send email notifications to the recipients you specify.

To configure email notifications:

The daemon notifies the recipients each time a health check run completes or when the daemon experiences a problem.

1. Specify a comma-delimited list of email addresses, as follows:

```
$ ./orachk -set  
"NOTIFICATION_EMAIL=some.person@acompany.com,another.person@acompany.com"
```

```
$ ./exachk -set  
"NOTIFICATION_EMAIL=some.person@acompany.com,another.person@acompany.com"
```

2. Test the email notification configuration using the `-testemail` option, as follows:

```
$ ./orachk -testemail all
```

```
$ ./exachk -testemail all
```

After the first health check run, the daemon notifies the recipients with report output attached.

For the subsequent health check runs after the first email notification, the daemon emails the summary of differences between the most recent runs to all recipients specified in the `NOTIFICATION_EMAIL` list.

2.5.3.4 collection_retention

Set the `collection_retention` daemon option to purge health check collection results that are older than a specified number of days.

To configure collection retention period:

1. Set the `collection_retention` option, as follows:

```
collection_retention=number_of_days
```

If you do not set this option, then the daemon does not purge the stale collection.

2. Set the `collection_retention` option to an appropriate number of days based on:
 - Frequency of your scheduled collections
 - Size of the collection results
 - Available disk space

For example:

```
$ ./orachk -set "collection_retention=60"
```

```
$ ./exachk -set "collection_retention=60"
```

2.5.3.5 PASSWORD_CHECK_INTERVAL

The `PASSWORD_CHECK_INTERVAL` daemon option defines the frequency, in hours, for the daemon to validate the passwords entered when the daemon was started the first time.

If an invalid password is found due to a password change, then the daemon stops, makes an entry in the daemon log, and then sends an email notification message to the recipients specified in the `NOTIFICATION_EMAIL` option.

To configure password validation frequency:

1. Set the `PASSWORD_CHECK_INTERVAL` option, as follows:

```
PASSWORD_CHECK_INTERVAL=number_of_hours
```

If you do not set the `PASSWORD_CHECK_INTERVAL` option, then the daemon cannot actively check password validity and fails the next time the daemon tries to run after a password change. Using the `PASSWORD_CHECK_INTERVAL` option enables you to take corrective action and restart the daemon with the correct password rather than having failed collections.

2. Set the `PASSWORD_CHECK_INTERVAL` option to an appropriate number of hours based on:
 - Frequency of your scheduled collections
 - Password change policies

For example:

```
$ ./orachk -set "PASSWORD_CHECK_INTERVAL=1"
```

```
$ ./exachk -set "PASSWORD_CHECK_INTERVAL=1"
```

2.5.3.6 Setting Multiple Option Profiles for the Daemon

Use only one daemon process for each server. Do not start a single daemon on multiple databases in a cluster, or multiple daemons on the same database.

The daemon does not start, if the daemon detects another Oracle ORAchk or Oracle EXAchk daemon process running locally.

Define multiple different run profiles using the same daemon. Defining multiple different run profiles enables you to run multiple different health checks with different daemon options, such as different schedules, email notifications, and automatic run flags. The daemon manages all profiles.

To set multiple option profiles for the daemon:

1. Define daemon option profiles using the `-id id` option before the `-set` option.

Where, `id` is the name of the profile

```
$ ./orachk -id id -set "option=value"
```

```
$ ./exachk -id id -set "option=value"
```

For example, if the database administrator wants to run checks within the `dba` profile and the system administrator wants to run checks in the `sysadmin` profile, then configure the daemon using the profiles option.

Define the database administrator profile as follows:

```
$ ./orachk -id dba -set "NOTIFICATION_EMAIL=dba@example.com;\
AUTORUN_SCHEDULE=4,8,12,16,20 * * * *;AUTORUN_FLAGS=-profile dba -tag dba;\
collection_retention=30"
```

```
Created notification_email for ID[dba]
Created autorun_schedule for ID[dba]
Created autorun_flags for ID[dba]
Created collection_retention for ID[dba]
```

```
$ ./exachk -id dba -set "NOTIFICATION_EMAIL=dba@example.com;\
AUTORUN_SCHEDULE=4,8,12,16,20 * * * *; AUTORUN_FLAGS=-profile dba -tag dba;\
collection_retention=30"
```

```
Created notification_email for ID[dba]
Created autorun_schedule for ID[dba]
Created autorun_flags for ID[dba]
Created collection_retention for ID[dba]
```

Define the system administrator profile as follows:

```
$ ./orachk -id sysadmin -set "NOTIFICATION_EMAIL=sysadmin@example.com;\
AUTORUN_SCHEDULE=3 * * 1,3,5; AUTORUN_FLAGS=-profile sysadmin -tag sysadmin;\
collection_retention=60"
```

```
Created notification_email for ID[sysadmin]
Created autorun_schedule for ID[sysadmin]
Created autorun_flags for ID[sysadmin]
Created collection_retention for ID[sysadmin]
```

```
$ ./exachk -id sysadmin -set "NOTIFICATION_EMAIL=sysadmin@example.com;\
AUTORUN_SCHEDULE=3 * * 1,3,5; AUTORUN_FLAGS=-profile sysadmin -tag sysadmin;\
collection_retention=60"
```

```
Created notification_email for ID[sysadmin]
Created autorun_schedule for ID[sysadmin]
Created autorun_flags for ID[sysadmin]
Created collection_retention for ID[sysadmin]
```

2.5.3.7 Getting Existing Options for the Daemon

Query the values that you set for the daemon options.

To query the values, use

```
[-id ID] -get option | all
```

where:

- *ID* is a daemon option profile
- *option* is a specific daemon option you want to retrieve
- *all* returns values of all options

To get existing options for the daemon:

1. To get a specific daemon option:

For example:

```
$ ./orachk -get NOTIFICATION_EMAIL
```

```
ID: orachk.default
-----
notification_email = some.body@example.com
```

```
$ ./exachk -get NOTIFICATION_EMAIL
```

```
ID: exachk.default
-----
notification_email = some.body@example.com
```

2. To query multiple daemon option profiles:

For example:

```
$ ./orachk -get NOTIFICATION_EMAIL

ID: orachk.default
-----
notification_email = some.body@example.com

ID: dba
-----
notification_email = dba@example.com

ID: sysadmin
-----
notification_email = sysadmin@example.com
```

```
$ ./exachk -get NOTIFICATION_EMAIL

ID: exachk.default
-----
notification_email = some.person@example.com

ID: dba
-----
notification_email = dba@example.com

ID: sysadmin
-----
notification_email = sysadmin@example.com
```

3. To limit the request to a specific daemon option profile, use the `-id ID -get option` option:

For example:

To get the `NOTIFICATION_EMAIL` for a daemon profile called `dba` :

```
$ ./orachk -id dba -get NOTIFICATION_EMAIL

ID: dba
-----
notification_email = dba@example.com

$ ./exachk -id dba -get NOTIFICATION_EMAIL

ID: dba
-----
notification_email = dba@example.com
```

4. To get all options set, use the `-get all` option:

For example:

```
$ ./orachk -get all

ID: orachk.default
-----
notification_email = some.body@example.com
autorun_schedule = 3 * * 0
collection_retention = 30
password_check_interval = 1
```

```
$ ./exachk -get all

ID: exachk.default
-----
notification_email = some.body@example.com
autorun_schedule = 3 * * 0
collection_retention = 30
password_check_interval = 1
```

5. To query all daemon option profiles:

For example:

```
$ ./orachk -get all

ID: orachk.default
-----
notification_email = some.body@example.com
autorun_schedule = 3 * * 0
collection_retention = 30
password_check_interval = 12
```

```
ID: dba
-----
notification_email = dba@example.com
autorun_schedule = 4,8,12,16,20 * * *
autorun_flags = -profile dba - tag dba
collection_retention = 30
password_check_interval = 1
```

```
ID: sysadmin
-----
notification_email = sysadmin@example.com
autorun_schedule = 3 * * 1,3,5
autorun_flags = -profile sysadmin -tag sysadmin
collection_retention = 60
password_check_interval = 1
```

```
$ ./exachk -get all

ID: exachk.default
-----
notification_email = some.body@example.com
autorun_schedule = 3 * * 0
collection_retention = 30
password_check_interval = 1
```

```
ID: dba
-----
notification_email = dba@example.com
autorun_schedule = 4,8,12,16,20 * * *
autorun_flags = -profile dba - tag dba
collection_retention = 30
password_check_interval = 1
```

```
ID: sysadmin
-----
notification_email = sysadmin@example.com
autorun_schedule = 3 * * 1,3,5
autorun_flags = -profile sysadmin -tag sysadmin
collection_retention = 60
password_check_interval = 1
```

6. To get all the options set for a daemon profile, for example, a daemon profile called dba:

```
$ ./orachk -id dba -get all

ID: dba
-----
notification_email = dba@example.com
autorun_schedule = 4,8,12,16,20 * * *
autorun_flags = -profile dba - tag dba
collection_retention = 30
password_check_interval = 1

$ ./exachk -id dba -get all

ID: dba
-----
notification_email = dba@example.com
autorun_schedule = 4,8,12,16,20 * * *
autorun_flags = -profile dba - tag dba
collection_retention = 30
password_check_interval = 1
```

2.5.4 Querying the Status and Next Planned Daemon Run

Query the status and next automatic run schedule of the running daemon.

```
-d status|info|nextautorun
```

- -d status: Checks if the daemon is running.
- -d info: Displays information about the running daemon.
- -d nextautorun [-id ID]: Displays the next automatic run time.

To query the status and next planned daemon run:

1. To check if the daemon is running, use -d status:

```
$ ./orachk -d status

$ ./exachk -d status
```

If the daemon is running, then the daemon confirms and displays the PID.

2. To query more detailed information about the daemon, use -d info:

```
$ ./orachk -d info

$ ./exachk -d info
```

The daemon responds with the following information:

- Node on which the daemon is installed
 - Version
 - Install location
 - Time when the daemon was started
3. To query the next scheduled health check run, use -d nextautorun:

```
$ ./orachk -d nextautorun
```

```
$ ./exachk -d nextautorun
```

The daemon responds with details of schedule.

If you have configured multiple daemon option profiles, then the output shows whichever is scheduled to run next.

If you have configured multiple daemon option profiles, then query the next scheduled health check run of a specific profile using `-id ID -d nextautorun`:

```
$ ./orachk -d ID -d nextautorun
```

```
$ ./exachk -d ID -d nextautorun
```

The daemon responds with details of the schedule for the daemon options profile ID you have specified.

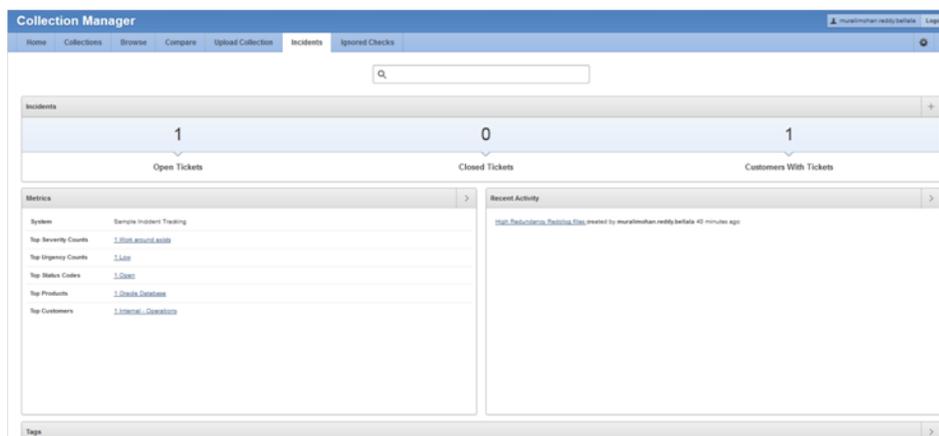
2.6 Tracking Support Incidents

The **Incidents** tab gives you a complete system for tracking support incidents.

- Specify contact details of each customer, products and categories, and then set up values to limit status codes, severity, and urgency attributes for an incident
- Raise a new ticket by clicking the Delta (Δ) symbol. Oracle Health Check Collections Manager displays the delta symbol only in the **Collections** and **Browse** tabs
- The **Browse** tab enables you to create a new ticket on individual checks
- The **Collections** tab enables you to create a single ticket for entire the collection
- Delta (Δ) symbol is color coded red, blue, and green based on the ticket status
 - **RED (No Incident ticket exists)**: Initiates the process to create a new incident ticket for the collection or individual checks
 - **BLUE (An open Incident ticket exists)**: Opens the incident ticket for editing
 - **GREEN (A closed Incident ticket exists)**: Opens the closed incident ticket for viewing
- Track the progress of the ticket in an update area of the ticket, or add attachments and links to the incident
- Use tags to classify incidents and use the resulting tag cloud in your reports
- Incident access and management happen only within your access control range

Note:

Incident Tracking feature is a basic stand-alone system and it is not designed for integration with other commercial enterprise-level trouble ticketing systems.

Figure 2-7 Incidents Tab**Incident Tracking Features**

- Search options
- Track and analyze incident tickets
- Flexible and updateable incident status
- Robust reporting
- Link, Note, and File Attachments
- Flexible Access Control (reader, contributor, administrator model)

Related Topics

- [Creating or Editing Incidents Tickets](#)
Create or edit incident tickets for individual checks or for an entire collection.
- *Oracle ORAchk and EXAchk User's Guide*

2.7 Tracking File Attribute Changes and Comparing Snapshots

Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.

Changes to the attributes of files such as owner, group, or permissions can cause unexpected consequences. Proactively monitor and mitigate the issues before your business gets impacted.

- [Using the File Attribute Check With the Daemon](#)
You must have Oracle Grid Infrastructure installed and running before you use `-fileattr`.
- [Taking File Attribute Snapshots](#)
By default, Oracle Grid Infrastructure homes and all the installed Oracle Database homes are included in the snapshots.
- [Including Directories to Check](#)
Include directories in the file attribute changes check.

- [Excluding Directories from Checks](#)
Exclude directories from file attribute changes checks.
- [Rechecking Changes](#)
Compare the new snapshot with the previous one to track changes.
- [Designating a Snapshot As a Baseline](#)
Designate a snapshot as a baseline to compare with other snapshots.
- [Restricting System Checks](#)
Restrict Oracle ORAchk and Oracle EXAchk to perform only file attribute changes checks.
- [Removing Snapshots](#)
Remove the snapshots diligently.

2.7.1 Using the File Attribute Check With the Daemon

You must have Oracle Grid Infrastructure installed and running before you use `fileattr`.

To use file attribute check with the daemon:

1. Start the daemon.

```
./orachk -d start
```

2. Start the client run with the `-fileattr` options.

```
./orachk -fileattr start -includedir "/root/myapp,/etc/oratab" -excludediscovery
./orachk -fileattr check -includedir "/root/myapp,/etc/oratab" -excludediscovery
```

3. Specify the output directory to store snapshots with the `-output` option.

```
./orachk -fileattr start -output "/tmp/mysnapshots"
```

4. Specify a descriptive name for the snapshot with the `-tag` option to identify your snapshots.

For example:

```
./orachk -fileattr start -tag "BeforeXYZChange"
Generated snapshot directory-
orachk_myserver65_20160329_052056_ BeforeXYZChange
```

2.7.2 Taking File Attribute Snapshots

By default, Oracle Grid Infrastructure homes and all the installed Oracle Database homes are included in the snapshots.

To take file attribute snapshots:

1. To start the first snapshot, run the `-fileattr start` command.

```
./orachk -fileattr start
```

```
./exachk -fileattr start
```

```
$ ./orachk -fileattr start
CRS stack is running and CRS_HOME is not set. Do you want to set CRS_HOME
to /u01/app/11.2.0.4/grid?[y/n][y]
Checking ssh user equivalency settings on all nodes in cluster
Node mysrv22 is configured for ssh user equivalency for oradb user
```

Node mysrv23 is configured for ssh user equivalency for oradb user

```
List of directories(recursive) for checking file attributes:
/u01/app/oradb/product/11.2.0/dbhome_11202
/u01/app/oradb/product/11.2.0/dbhome_11203
/u01/app/oradb/product/11.2.0/dbhome_11204
orachk has taken snapshot of file attributes for above directories at: /orahome/
oradb/orachk/orachk_mysrv21_20160504_041214
```

2.7.3 Including Directories to Check

Include directories in the file attribute changes check.

To include directories to check:

1. Run the file attribute changes check command with the `-includedir directories` option.

Where, *directories* is a comma-delimited list of directories to include in the check.

For example:

```
./orachk -fileattr start -includedir "/home/oradb,/etc/oratab"

./exachk -fileattr start -includedir "/home/oradb,/etc/oratab"

$ ./orachk -fileattr start -includedir "/root/myapp/config/"
CRS stack is running and CRS_HOME is not set. Do you want to set CRS_HOME
to /u01/app/12.2.0/grid?[y/n][y]
Checking for prompts on myserver18 for oragrid user...
Checking ssh user equivalency settings on all nodes in cluster
Node myserver17 is configured for ssh user equivalency for root user
List of directories(recursive) for checking file attributes:
/u01/app/12.2.0/grid
/u01/app/oradb/product/12.2.0/dbhome_1
/u01/app/oradb2/product/12.2.0/dbhome_1
/root/myapp/config/
orachk has taken snapshot of file attributes for above directories at: /root/orachk/
orachk_myserver18_20160511_032034
```

2.7.4 Excluding Directories from Checks

Exclude directories from file attribute changes checks.

To exclude directories from checks:

1. Run the file attribute changes check command to exclude directories that you do not list in the `-includedir` discover list by using the `-excludediscovery` option.

For example:

```
$ ./orachk -fileattr start -includedir "/root/myapp/config/" -excludediscovery
CRS stack is running and CRS_HOME is not set. Do you want to set CRS_HOME
to /u01/app/12.2.0/grid?[y/n][y]
Checking for prompts on myserver18 for oragrid user...
Checking ssh user equivalency settings on all nodes in cluster
Node myserver17 is configured for ssh user equivalency for root user
List of directories(recursive) for checking file attributes:
/root/myapp/config/
orachk has taken snapshot of file attributes for above directories at: /root/orachk/
orachk_myserver18_20160511_032209
```

2.7.5 Rechecking Changes

Compare the new snapshot with the previous one to track changes.

To recheck changes:

1. Run the file attribute changes check command with the `check` option to take a new snapshot, and run a normal health check collection.

The `-fileattr check` command compares the new snapshot with the previous snapshot.

For example:

```
./orachk -fileattr check
```

```
./exachk -fileattr check
```

Note:

To obtain an accurate comparison between the snapshots, you must use `-fileattr check` with the same options that you used with the previous snapshot collection that you obtained with `-fileattr start`.

For example, if you obtained your first snapshot by using the options `-includedir "/somedir" -excludediscovery` when you ran `-fileattr start`, then you must include the same options with `-fileattr check` to obtain an accurate comparison.

```
$ ./orachk -fileattr check -includedir "/root/myapp/config" -excludediscovery
CRS stack is running and CRS_HOME is not set. Do you want to set CRS_HOME
to /u01/app/12.2.0/grid?[y/n][y]
Checking for prompts on myserver18 for oragrid user...
Checking ssh user equivalency settings on all nodes in cluster
Node myserver17 is configured for ssh user equivalency for root user
List of directories(recursive) for checking file attributes:
/root/myapp/config
Checking file attribute changes...
.
"/root/myapp/config/myappconfig.xml" is different:
Baseline :      0644   oracle      root /root/myapp/config/myappconfig.xml
Current  :      0644    root      root /root/myapp/config/myappconfig.xml
...
```

Results of the file attribute changes are reflected in the **File Attribute Changes** section of the HTML output report.

2.7.6 Designating a Snapshot As a Baseline

Designate a snapshot as a baseline to compare with other snapshots.

To designate a snapshot as a baseline:

1. Run the file attribute changes check command with the `-baseline path_to_snapshot` option.

The `-baseline path_to_snapshot` command compares a specific baseline snapshot with other snapshots, if you have multiple different baselines to check.

```
./orachk -fileattr check -baseline path_to_snapshot
```

```
./exachk -fileattr check -baseline path_to_snapshot
```

For example:

```
./orachk -fileattr check -baseline "/tmp/Snapshot"
```

2.7.7 Restricting System Checks

Restrict Oracle ORAchk and Oracle EXAchk to perform only file attribute changes checks.

By default, `-fileattr check` also performs a full health check run.

To restrict system checks:

1. Run the file attribute changes check command with the `-fileattronly` option.

```
./orachk -fileattr check -fileattronly
```

```
./exachk -fileattr check -fileattronly
```

2.7.8 Removing Snapshots

Remove the snapshots diligently.

To remove snapshots:

1. Run the file attribute changes check command with the `remove` option:

```
./orachk -fileattr remove
```

```
./exachk -fileattr remove
```

For example:

```
$ ./orachk -fileattr remove
CRS stack is running and CRS_HOME is not set. Do you want to set CRS_HOME
to /u01/app/12.2.0/grid?[y/n][y]
Checking for prompts on myserver18 for oragrid user...
Checking ssh user equivalency settings on all nodes in cluster
Node myserver17 is configured for ssh user equivalency for root user

List of directories(recursive) for checking file attributes:
/u01/app/12.2.0/grid
/u01/app/oradb/product/12.2.0/dbhome_1
/u01/app/oradb2/product/12.2.0/dbhome_1
Removing file attribute related files...
...
```

2.8 Collecting and Consuming Health Check Data

Oracle Health Check Collections Manager for Oracle Application Express 4.2 provides you an enterprise-wide view of your health check collection data.

- [Selectively Capturing Users During Login](#)
Configure Oracle Health Check Collections Manager to capture user details and assign the users Oracle Health Check Collections Manager roles.
- [Bulk Mapping Systems to Business Units](#)
Oracle Health Check Collections Manager provides an XML bulk upload option so that you can quickly map many systems to business units.
- [Adjusting or Disabling Old Collections Purging](#)
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.
- [Uploading Collections Automatically](#)
Configure Oracle ORAchk and Oracle EXAchk to upload check results automatically to the Oracle Health Check Collections Manager database.
- [Viewing and Reattempting Failed Uploads](#)
Configure Oracle ORAchk and Oracle EXAchk to display and reattempt to upload the failed uploads.
- [Authoring User-Defined Checks](#)
Define, test, and maintain your own checks that are specific to your environment.
- [Finding Which Checks Require Privileged Users](#)
Use the **Privileged User** filter in the Health Check Catalogs to find health checks that must be run by privileged users, such as `root`.
- [Creating or Editing Incidents Tickets](#)
Create or edit incident tickets for individual checks or for an entire collection.
- [Viewing Clusterwide Linux Operating System Health Check \(VMPScan\)](#)
On Linux systems, view a summary of the VMPScan report in the Clusterwide Linux Operating System Health Check (VMPScan) section of the Health Check report.

Related Topics

- *Oracle ORAchk and EXAchk User's Guide*

2.8.1 Selectively Capturing Users During Login

Configure Oracle Health Check Collections Manager to capture user details and assign the users Oracle Health Check Collections Manager roles.

Automatically capturing users during login automates user management. You need not create users manually.

By default, Oracle Health Check Collections Manager:

- Captures details of users that are logging in with LDAP authentication
- Assigns them Oracle Health Check Collections Manager roles, for example, DBA role.

Note:

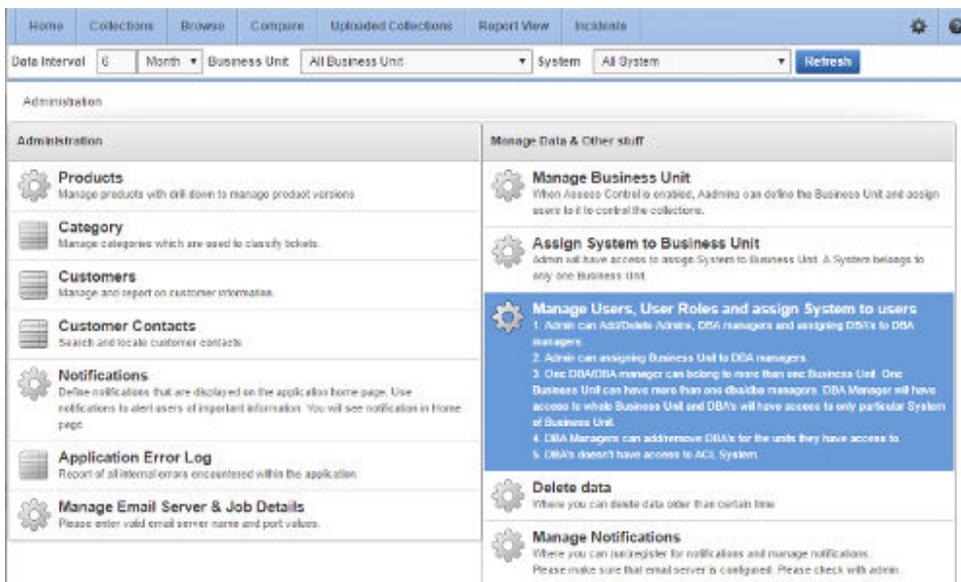
The Oracle Health Check Collections Manager roles are specific to Oracle Health Check Collections Manager and do not equate to system privileges. For example, the DBA role is not granted SYSDBA system privilege.

However, you can disable automatic capture and re-enable anytime later. If you disable, then you must manually create users and assign them roles.

To enable or disable capturing user details automatically:

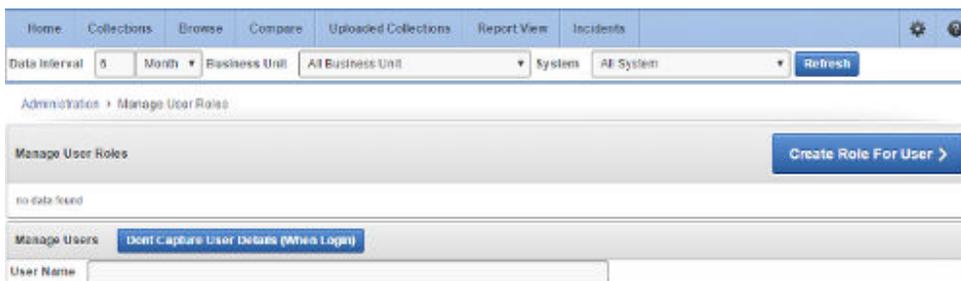
1. Click **Administration**, and then select **Manage Users, User Roles and assign System to users**.

Figure 2-8 Manage Users, User Roles and assign System to users



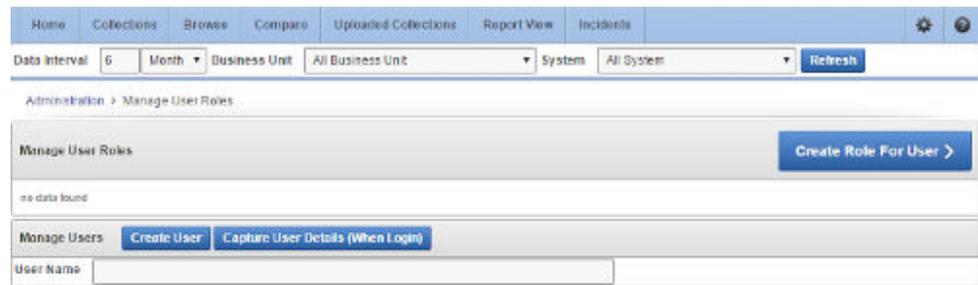
2. To disable automatic capture of users details, click **Don't Capture User Details (When Login)**.

Figure 2-9 Don't Capture User Details (When Login)



3. To re-enable automatic capture of user details, click **Capture User Details (When Login)**.

Figure 2-10 Capture User Details (When Login)



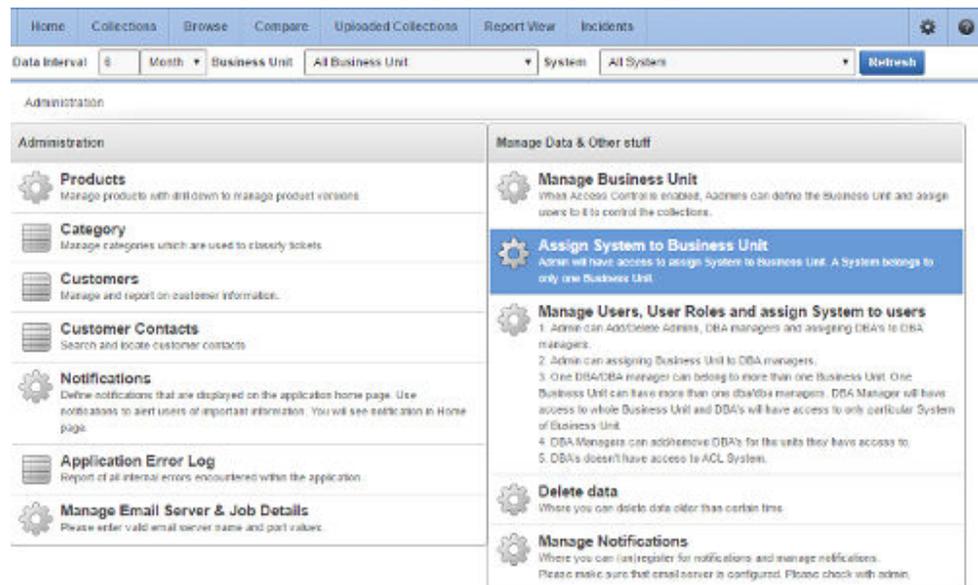
2.8.2 Bulk Mapping Systems to Business Units

Oracle Health Check Collections Manager provides an XML bulk upload option so that you can quickly map many systems to business units.

To bulk map systems to the business units:

1. Click **Administration**, then select **Assign System to Business Unit**.

Figure 2-11 Assign System to Business Unit



2. Click **Bulk Mapping**.

Figure 2-12 Bulk Mapping

Unit Name	System Name	Host Names
BLUEWIDGETSALES	mysystem1	host1 host2 host3 host4
BLUEWIDGETDEVELOPMENT	mysystem2	host5 host6
SAMPLE	sample	obtain10

3. Upload a mapping XML.
 - a. Click **Generate XML File (Current Mapping)**.
 - b. Download the resulting XML file that contains your current system to business unit mappings.

Figure 2-13 Upload a mapping XML

File Name	Uploaded By	Uploaded On	Delete XML
Download	DATE_GENERATED	UPLOADED_BY	
	05-MAY-16 06:05:27 000000 AM	KAVITHA.DIHANASEKAR	

- c. Amend the XML to show mappings that you want.
- d. Upload new Mapping XML through **Upload Mapping (XML File)**.

2.8.3 Adjusting or Disabling Old Collections Purging

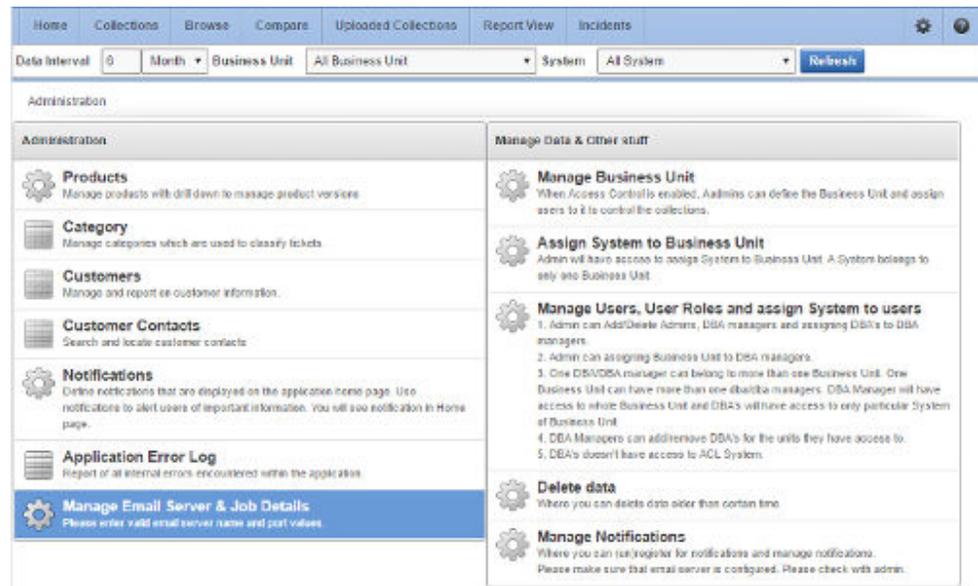
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.

By default, Oracle Health Check Collections Manager purges collections older than three months.

To adjust or disable the collection purging frequency:

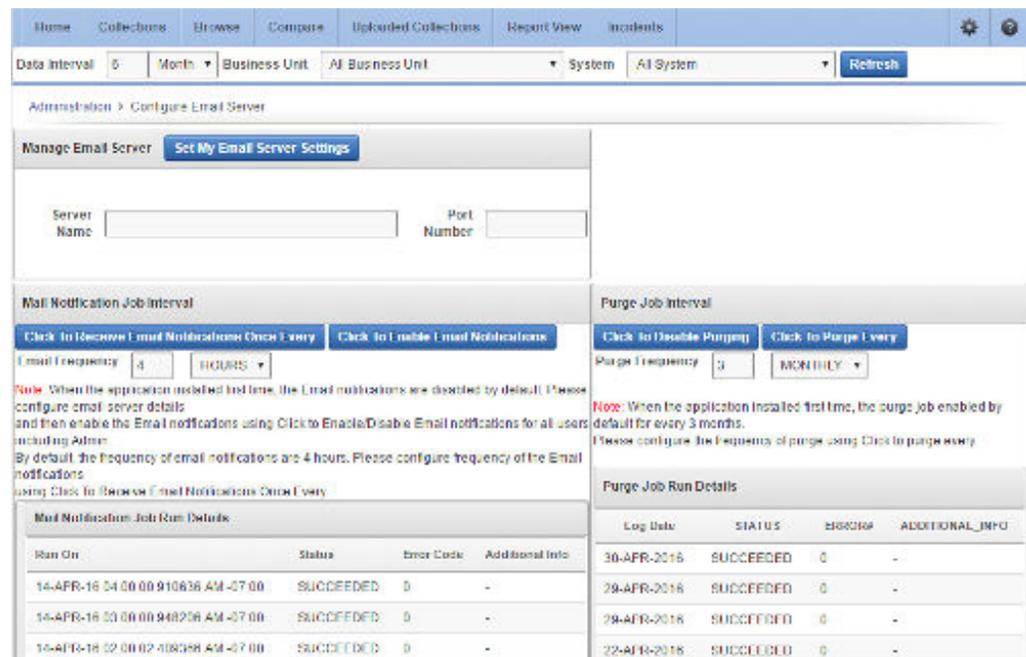
1. Click **Administration**, and then select **Manage Email Server & Job Details**.

Figure 2-14 Manage Email Server and Job Details



2. Select an appropriate option:
 - Change the frequency of purges by setting different values in **Purge Frequency** . Then click **Click To Purge Every**.
 - To disable purging, click **Click To Disable Purging**.
 - To re-enable purging, click **Click To Enable Purging**.

Figure 2-15 Configure Purging



2.8.4 Uploading Collections Automatically

Configure Oracle ORAchk and Oracle EXAchk to upload check results automatically to the Oracle Health Check Collections Manager database.

Specify the connection string and the password to connect to the database. Oracle Health Check Collections Manager stores the connection details in an encrypted wallet.

To configure Oracle ORAchk and Oracle EXAchk to upload check results automatically:

1. Specify the connection details using the `-setdbupload` option. For default options, use `-setdbupload all`.

```
orachk -setdbupload all
```

```
exachk -setdbupload all
```

Oracle Health Check Collections Manager prompts you to enter the values for the connection string and password. Oracle Health Check Collections Manager stores these values in an encrypted wallet file.

2. Verify the values set in the wallet, using the `-getdbupload` option.

```
orachk -getdbupload
```

```
exachk -getdbupload
```

Oracle ORAchk and Oracle EXAchk automatically use the default values set in the `RAT_UPLOAD_USER` and `RAT_ZIP_UPLOAD_TABLE` environment variables.

3. Verify, using the `-checkdbupload` option if Oracle ORAchk and Oracle EXAchk successfully connect to the database.

```
orachk -checkdbupload
```

```
exachk -checkdbupload
```

4. Set database uploads for Oracle ORAchk and Oracle EXAchk check results.

```
orachk -setdbupload all
```

```
exachk -setdbupload all
```

Note:

Use fully qualified address for the connect string as mentioned in the previous example. Do not use an alias from the `tnsnames.ora` file.

Using fully qualified address eliminates the need to rely on `tnsnames.ora` file name resolution on all the servers where you run the tool.

5. Review Oracle ORAchk and Oracle EXAchk database check result uploads.

```
orachk -getdbupload
```

```
exachk -getdbupload
```

Example 2-1 Checking Oracle ORAchk and Oracle EXAchk Check Result Uploads

```
$ ./orachk -checkdbupload
```

Configuration is good to upload result to database.

At the end of health check collection, Oracle ORAchk and Oracle EXAchk check if the required connection details are set (in the wallet or the environment variables). If the connection details are set properly, then Oracle ORAchk and Oracle EXAchk upload the collection results.

To configure many Oracle ORAchk and Oracle EXAchk instances:

1. Create the wallet once with the `-setdbupload all` option, then enter the values when prompted.
2. Copy the resulting wallet directory to each Oracle ORAchk and Oracle EXAchk instance directories.

You can also set the environment variable `RAT_WALLET_LOC` to point to the location of the wallet directory.

Other configurable upload values are:

- `RAT_UPLOAD_USER`: Controls which user to connect as (default is `ORACHKCM`).
- `RAT_UPLOAD_TABLE`: Controls the table name to store non-zipped collection results in (not used by default).
- `RAT_PATCH_UPLOAD_TABLE`: Controls the table name to store non-zipped patch results in (not used by default).
- `RAT_UPLOAD_ORACLE_HOME`: Controls `ORACLE_HOME` used while establishing connection and uploading.

By default, the `ORACLE_HOME` environment variable is set to the Oracle Grid Infrastructure Grid home that Oracle ORAchk and Oracle EXAchk discover.

`RCA13_DOCS`: Not configurable to use Oracle Health Check Collections Manager because `RCA13_DOCS` is the table Oracle Health Check Collections Manager looks for.

`RAT_UPLOAD_TABLE` and `RAT_PATCH_UPLOAD_TABLE`: Not used by default because the zipped collection details are stored in `RCA13_DOCS`.

Configure `RAT_UPLOAD_TABLE` and `RAT_PATCH_UPLOAD_TABLE` environments variables if you are using your own custom application to view the collection results.

You can also set these values in the wallet.

For example:

```
$ ./orachk -setdbupload all
```

```
$ ./exachk -setdbupload all
```

This prompts you for and set the `RAT_UPLOAD_CONNECT_STRING` and `RAT_UPLOAD_PASSWORD`, then use

```
$ ./orachk -setdbupload RAT_PATCH_UPLOAD_TABLE,RAT_PATCH_UPLOAD_TABLE
```

```
$ ./exachk -setdbupload RAT_PATCH_UPLOAD_TABLE,RAT_PATCH_UPLOAD_TABLE
```

 **Note:**

Alternatively, set all values set in the wallet using the environment variables. If you set the values using the environment variable `RAT_UPLOAD_CONNECT_STRING`, then enclose the values in double quotes.

For example:

```
export RAT_UPLOAD_CONNECT_STRING="(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)
(HOST=myserver44.example.com)(PORT=1521))(CONNECT_DATA=(SERVER=DEDICATED)
(SERVICE_NAME=orachkcm.example.com)))"
```

2.8.5 Viewing and Reattempting Failed Uploads

Configure Oracle ORAchk and Oracle EXAchk to display and reattempt to upload the failed uploads.

The tools store the values in the `collection_dir/outfiles/check_env.out` file to record if the previous database upload was successful or not.

The following example shows that database upload has been set up, but the last upload was unsuccessful:

```
DATABASE_UPLOAD_SETUP=1
DATABASE_UPLOAD_STATUS=0
```

To view and reattempt failed uploads:

1. To view failed collections, use the `-checkfaileduploads` option.

```
./orachk -checkfaileduploads
```

```
./exachk -checkfaileduploads
```

For example:

```
$ ./orachk -checkfaileduploads
```

```
List of failed upload collections
/home/oracle/orachk_myserver_042016_232011.zip
/home/oracle/orachk_myserver_042016_231732.zip
/home/oracle/orachk_myserver_042016_230811.zip
/home/oracle/orachk_myserver_042016_222227.zip
/home/oracle/orachk_myserver_042016_222043.zip
```

2. To reattempt collection upload, use the `-uploadfailed` option

Specify either all to upload all collections or a comma-delimited list of collections:

```
./orachk -uploadfailed all|list of failed collections
```

```
./exachk -uploadfailed all|list of failed collections
```

For example:

```
./orachk -uploadfailed "/home/oracle/orachk_myserver_042016_232011.zip, /
home/oracle/orachk_myserver_042016_231732.zip"
```



Note:

You cannot upload collections uploaded earlier because of the SQL unique constraint.

2.8.6 Authoring User-Defined Checks

Define, test, and maintain your own checks that are specific to your environment.

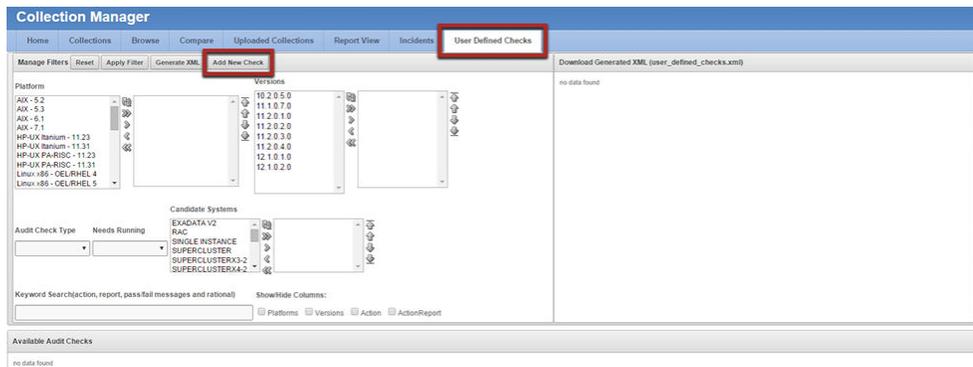
Oracle supports the framework for creating and running user-defined checks, but not the logic of the checks. It is your responsibility to test, verify, author, maintain, and support user-defined checks. At runtime, Oracle ORAchk and Oracle EXAchk script run the user-defined checks and display the results in the **User Defined Checks** section of the HTML report.

The user-defined checks are stored in the Oracle Health Check Collections Manager schema and output to an XML file, which is co-located with the ORAchk script. When run on your system, ORAchk 12.1.0.2.5 and later tries to find the XML file. If found, then Oracle ORAchk runs the checks contained therein and includes the results in the standard HTML report.

To author user-defined checks:

1. Click the **User Defined Checks** tab, then select **Add New Check**.

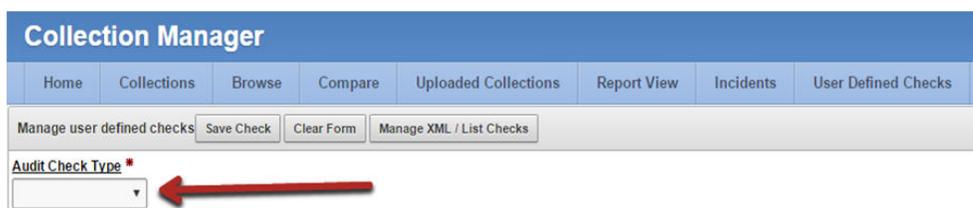
Figure 2-16 User-Defined Checks Tab



2. Select **OS Check** or **SQL Check** as **Audit Check Type**.

Operating system checks use a system command to determine the check status. SQL checks run an SQL statement to determine the check status.

Figure 2-17 User-Defined Checks Tab - Audit Check Type



Once you have selected an **Audit Check Type**, Oracle Health Check Collections Manager updates the applicable fields.

Any time during authoring, click the title of a field to see help documentation specific to that field.

Operating system and SQL commands are supported. Running user-defined checks as `root` is **NOT** supported.

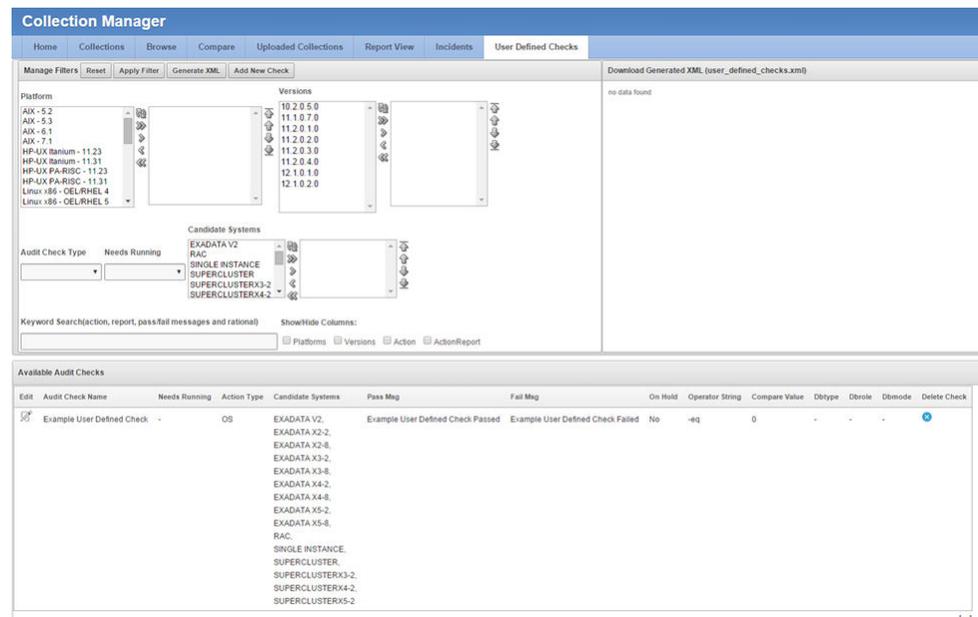
Figure 2-18 User-Defined Checks Tab - Audit Check Type - OS Check

The screenshot shows the 'Collection Manager' interface for 'User Defined Checks'. The 'Audit Check Type' is set to 'On Hold' and 'OS Check' is set to 'NO'. The 'Audit Check Name' field is empty. The 'OS Command' and 'OS Command for report' fields are also empty. The 'Oracle Version' dropdown is open, showing options: 10.2.0.5.0, 11.1.0.7.0, 11.2.0.1.0, 11.2.0.2.0, 11.2.0.3.0, and 11.2.0.4.0. The 'Platforms' dropdown is also open, showing options: AIX - 5.2, AIX - 5.3, AIX - 6.1, AIX - 7.1, HP-UX Itanium - 11.23, and HP-UX Itanium - 11.31. The 'Candidate Systems' list includes EXADATA V2, RAC, SINGLE INSTANCE, SUPERCLUSTER, SUPERCLUSTERX3-2, and SUPERCLUSTERX4-2. The 'Comparison Operator' and 'Comparison Value' fields are empty. The 'Alert Level' dropdown is set to '--Select AlertLevel--'. The 'Success Message (If Check Passes)' and 'Failure Message (If Check Fails)' fields are empty. The 'Benefit Impact', 'Risk', 'Action/Repair', 'User Comments', and 'Documents/Notes' fields are also empty. A red arrow points to the 'Oracle Version' dropdown with the text 'Click field titles for help'.

Once a check is created, the check is listed in the **Available Audit Checks** section.

Filter the checks using the filters on this page.

Figure 2-19 User-Defined Checks Tab - Available Audit Checks



3. Click the **Generate XML**.

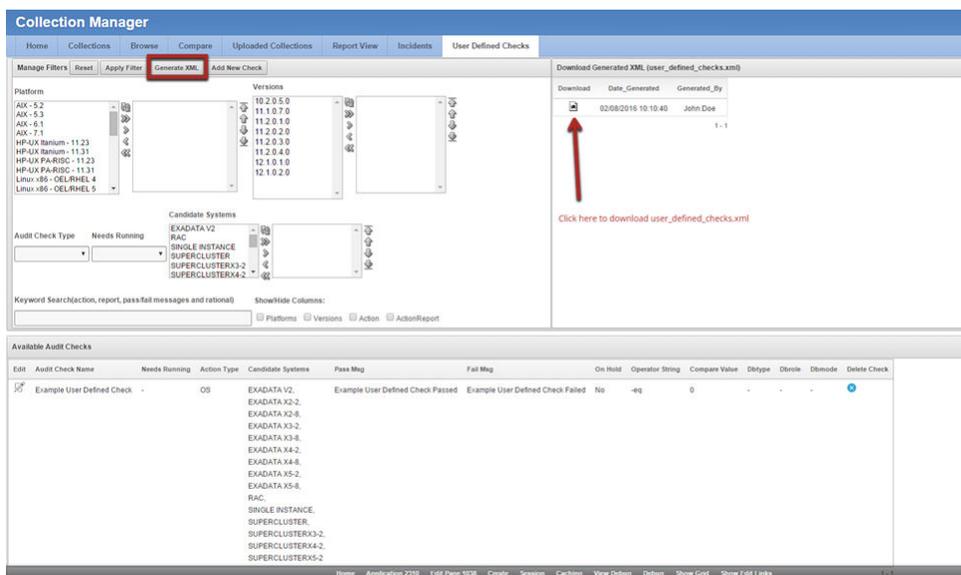
On the right, find a link to download the generated `user_defined_checks.xml` file.

The generated XML file includes all the checks that have been authored and have not been placed on hold. Placing checks on hold is equivalent to a logical delete. If there is a problem with a check or the logic is not perfect, then place the check on hold. The check that is placed on hold is not included in the XML file. If the check is production ready, then remove the hold to include the check the next time the XML file is generated.

4. Download and save the `user_defined_checks.xml` file into the same directory as the Oracle ORAchk and Oracle EXAchk tools.

Oracle ORAchk and Oracle EXAchk run the user-defined checks the next time they run.

Figure 2-20 User-Defined Checks Tab - Download User-Defined Checks



- Alternatively, to run only the user-defined checks use the profile `user_defined_checks`.

When this option is used, then the user-defined checks are the only checks run and the **User Defined Checks** section is the only one with results displayed in the report.

```
./orachk -profile user_defined_checks
```

```
./exachk -profile user_defined_checks
```

- To omit the user-defined checks at runtime, use the `-excludeprofile` option.

```
./orachk -excludeprofile user_defined_checks
```

```
./exachk -excludeprofile user_defined_checks
```

2.8.7 Finding Which Checks Require Privileged Users

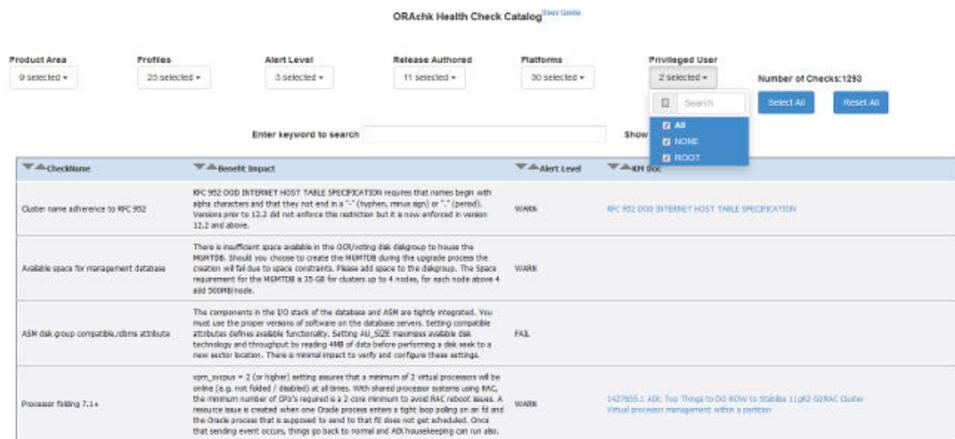
Use the **Privileged User** filter in the Health Check Catalogs to find health checks that must be run by privileged users, such as `root`.

Enable Javascript before you view the Health Check Catalogs.

To filter health checks by privileged users:

- Go to My Oracle Support note 1268927.2.
- Click the **Health Check Catalog** tab.
- Click **Open ORAchk Health Check Catalog** to open or download the `ORAchk_Health_Check_Catalog.html` file.
- Click the **Privileged User** drop-down list and then clear or select the check boxes appropriately.

Figure 2-21 Oracle ORAchk - Privileged User



Related Topics

- <https://support.oracle.com/rs?type=doc&id=1268927.2>

2.8.8 Creating or Editing Incidents Tickets

Create or edit incident tickets for individual checks or for an entire collection.

Oracle Health Check Collections Manager represents the statuses of each ticket with different colored icons. To act upon the tickets, click the icons.

- [Creating Incident Tickets](#)
- [Editing Incident Tickets](#)

2.8.8.1 Creating Incident Tickets

To create incident tickets:

1. Click the **Delta (Δ)** symbol colored RED.
2. Add your ticket details.
3. Click **Next**.
4. Select the **Product** and **Product Version**.
5. Click **Next**.
6. Select the **Urgency** of the ticket.
7. Select the **Severity** of the ticket.
8. Select the **Status** of the ticket.
9. Select the **Category** of the ticket.
10. Enter a summary and description of the incident.
11. Click **Create Ticket**.

Editing Incident Tickets

To edit incident tickets:

1. Click the **Incident** tab.
2. Click **Open Tickets**.
3. Click the ticket.
4. Click **Edit Ticket**.
5. Alter required details, click **Apply Changes**.



Note:

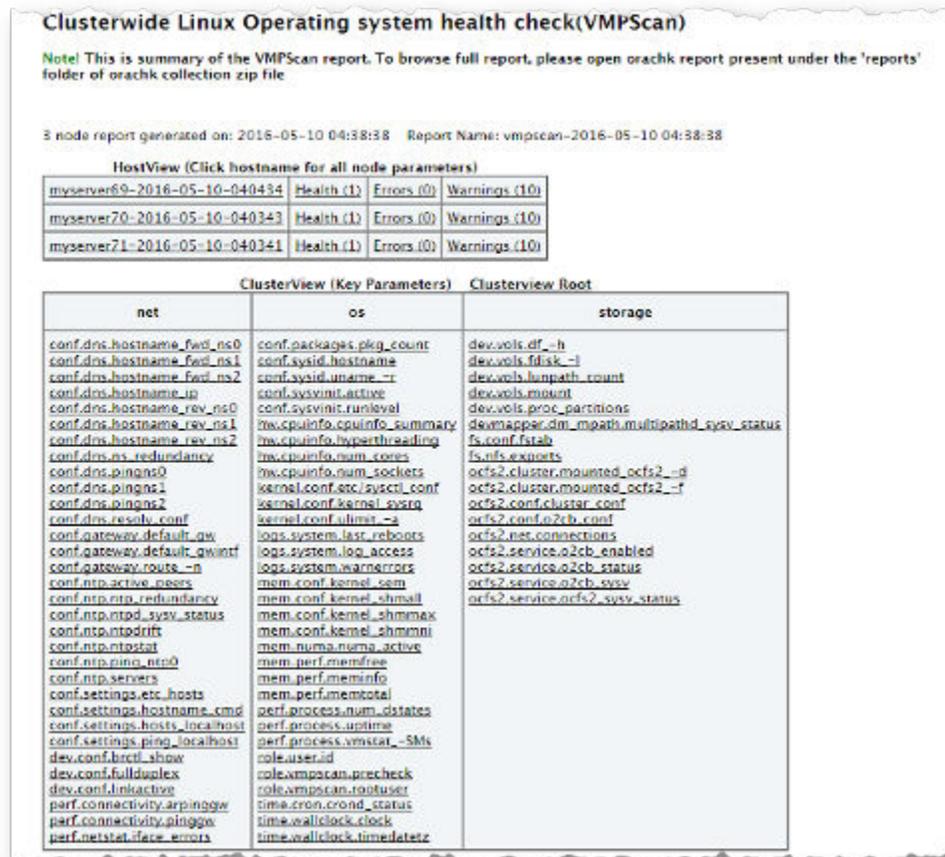
Click the delta symbol colored GREEN in the **Collections** or **Browse** tabs to edit incident tickets.

2.8.9 Viewing Clusterwide Linux Operating System Health Check (VMPScan)

On Linux systems, view a summary of the VMPScan report in the Clusterwide Linux Operating System Health Check (VMPScan) section of the Health Check report.

The full VMPScan report is also available within the `collection/reports` and `collection/outfiles/vmpscan` directory.

Figure 2-22 Clusterwide Linux Operating System Health Check (VMPScan)



Note:
The VMPScan report is included only when Oracle ORAchK is run on Linux systems.

2.9 Locking and Unlocking Storage Server Cells

Beginning with version 12.1.0.2.7, use Oracle EXAchK to lock and unlock storage server cells.

On the database server, if you configure passwordless SSH equivalency for the user that launched Oracle EXAchK to the root user on each storage server, then Oracle EXAchK uses SSH equivalency to complete the storage server checks. Run Oracle EXAchK from the Oracle Exadata storage server, if there is no SSH connectivity from the database to the storage server.

To lock and unlock cells, use the -unlockcells and -lockcells options for Oracle Exadata, Oracle SuperCluster and Zero Data Loss Recovery Appliance.

```
./exachk -unlockcells all | -cells [comma-delimited list of cell names or cell IPs]  
./exachk -lockcells all | -cells [comma-delimited list of cell names or cell IPs]
```

2.10 Integrating Health Check Results with Other Tools

Integrate health check results from Oracle ORAchk and Oracle EXAchk into Enterprise Manager and other third-party tools.

- [Integrating Health Check Results with Oracle Enterprise Manager](#)
Integrate health check results from Oracle ORAchk and Oracle EXAchk into Oracle Enterprise Manager.
- [Integrating Health Check Results with Third-Party Tool](#)
Integrate health check results from Oracle ORAchk and Oracle EXAchk into various third-party log monitoring and analytics tools, such as Elasticsearch and Kibana.
- [Integrating Health Check Results with Custom Application](#)
Oracle ORAchk and Oracle EXAchk upload collection results from multiple instances into a single database for easier consumption of check results across your enterprise.

2.10.1 Integrating Health Check Results with Oracle Enterprise Manager

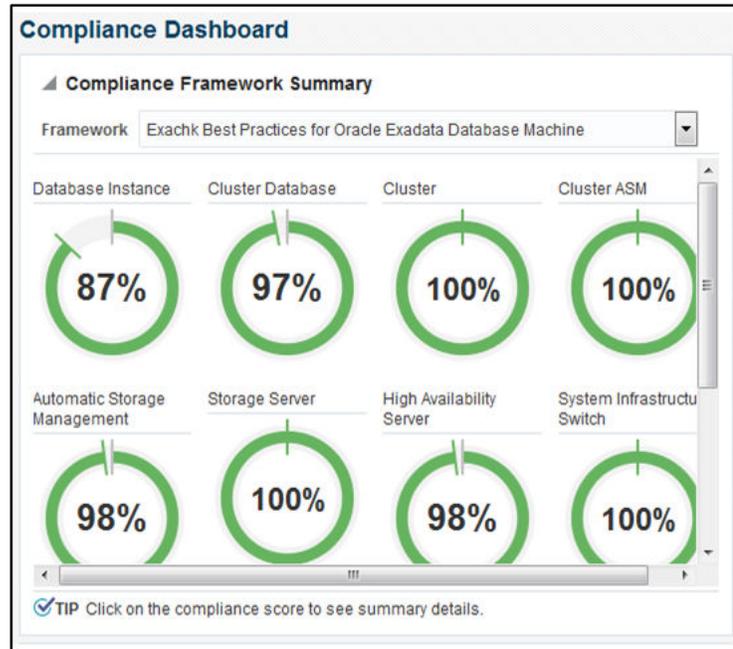
Integrate health check results from Oracle ORAchk and Oracle EXAchk into Oracle Enterprise Manager.

Oracle Enterprise Manager Cloud Control releases 13.1 and 13.2 support integration with Oracle ORAchk and Oracle EXAchk through the Oracle Enterprise Manager ORAchk Healthchecks Plug-in. The Oracle Engineered System Healthchecks plug-in supported integration with EXAchk for Oracle Enterprise Manager Cloud Control 12c release 12.1.0.5 and earlier releases.

With Oracle Enterprise Manager Cloud Control 13.1, Oracle ORAchk and Oracle EXAchk check results are integrated into the compliance framework. Integrating check results into the compliance framework enables you to display Compliance Framework Dashboards and browse checks by compliance standards.

- Integrate check results into Oracle Enterprise Manager compliance framework.
- View health check results in native Oracle Enterprise Manager compliance dashboards.

Figure 2-23 Compliance Dashboard



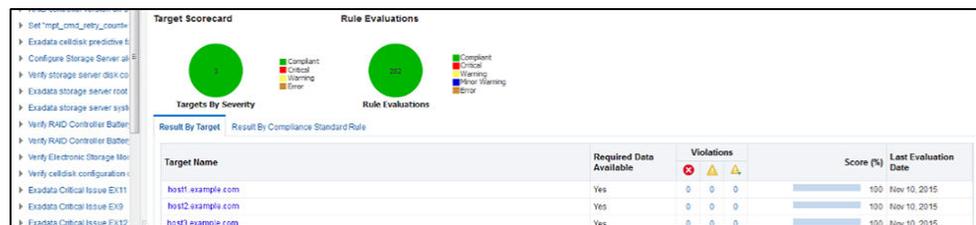
- Related checks are grouped into compliance standards where you can view targets checked, violations, and average score.

Figure 2-24 Compliance Standards

Compliance Standards	Applicable To	Compliance Standard State	Target Evaluations			Violations			Average Score (%)
			⊖	⚠	✔	⊖	⚠	✔	
Exachk Cluster ASM Best Practices For Oracle Exadata Database Machine	Cluster ASM	Production	0	0	1	0	0	0	100
Exachk Oracle Exadata Storage Server Best Practices For Oracle Exadata Database Machine	Oracle Exadata Storage Server	Production	0	0	3	0	0	0	100
Exachk System Infrastructure Switch Best Practices For Oracle Exadata Database Machine	System Infrastructure Switch	Production	0	0	3	0	0	0	100
Exachk Cluster Best Practices For Oracle Exadata Database Machine	Cluster	Production	0	0	1	0	0	0	100
Exachk Host Best Practices For Oracle Exadata Database Machine	Host	Production	0	0	2	2	2	13	93
Exachk Automatic Storage Management Best Practices For Oracle Exadata Database Machine	Automatic Storage Management	Production	0	0	2	2	1	0	97
Exachk Cluster Database Best Practices For Oracle Exadata Database Machine	Cluster Database	Production	0	0	1	5	3	1	97
Exachk Oracle High Availability Service Best Practices For Oracle Exadata Database Machine	Oracle High Availability Service	Production	0	0	2	2	0	0	98
Exachk Database Instance Best Practices For Oracle Exadata Database Machine	Database Instance	Production	0	0	2	32	8	0	87

- From within a compliance standard, drill-down to see individual check results and break the results by targets.

Figure 2-25 Compliance Standards Drill-Down





Note:

Although Oracle ORAchk and Oracle EXAchk do not require additional licenses, you require applicable Oracle Enterprise Manager licenses.

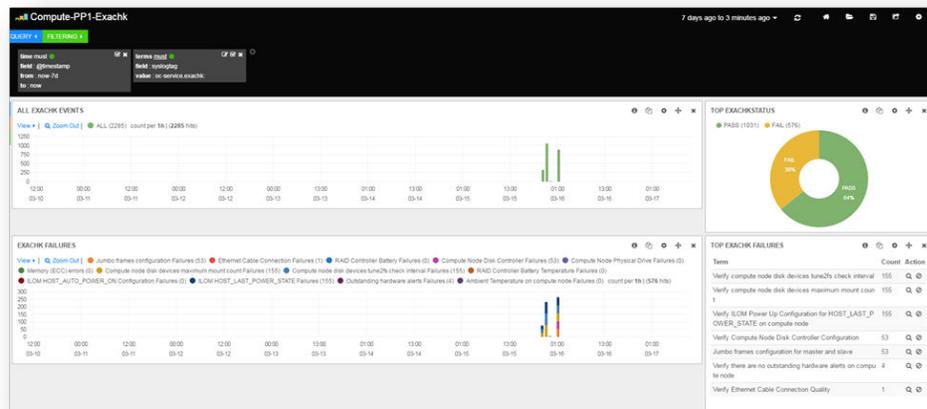
Related Topics

- Oracle Enterprise Manager ORAchk Healthchecks Plug-in User's Guide
- Oracle Enterprise Manager Licensing Information User Manual

2.10.2 Integrating Health Check Results with Third-Party Tool

Integrate health check results from Oracle ORAchk and Oracle EXAchk into various third-party log monitoring and analytics tools, such as Elasticsearch and Kibana.

Figure 2-26 Third-Party Tool Integration



Oracle ORAchk and Oracle EXAchk create JSON output results in the output upload directory, for example:

```
Report_Output_Dir/upload/mymachine_orachk_results.json
Report_Output_Dir/upload/mymachine_orachk_exceptions.json
```

```
Report_Output_Dir/upload/mymachine_exachk_results.json
Report_Output_Dir/upload/mymachine_exachk_exceptions.json
```

1. Run the `-syslog` option to write JSON results to the `syslog` daemon.

For example:

```
./orachk -syslog
```

```
./exachk -syslog
```

2. Verify the `syslog` configuration by running the following commands:

Oracle ORAchk and Oracle EXAchk use the message levels: CRIT, ERR, WARN, and INFO.

```
$ logger -p user.crit crit_message
$ logger -p user.err err_message
$ logger -p user.warn warn_message
$ logger -p user.info info_message
```

3. Verify in your configured message location, for example, /var/adm/messages that each test message is written.

Related Topics

- [Elasticsearch: RESTful, Distributed Search & Analytics | Elastic](#)
- [Kibana: Explore, Visualize, Discover Data | Elastic](#)
- [Logging Alerts to the syslogd Daemon](#)

2.10.3 Integrating Health Check Results with Custom Application

Oracle ORAchk and Oracle EXAchk upload collection results from multiple instances into a single database for easier consumption of check results across your enterprise.

Use Oracle Health Check Collections Manager or your own custom application to consume health check results.

1. Upload the collection results into the following tables at the end of a collection:

Table 2-3 Uploading Collection Results into a Database

Table	What Get's Uploaded
rca13_docs	Full zipped collection results.
auditcheck_result	Health check results.
auditcheck_patch_resu lt	Patch check results.

If you install Oracle Health Check Collections Manager, then these tables are created by the install script.

2. If the tables are not created, then use the following DDL statements:

- **DDL for the RCA13_DOCS table**

```
CREATE TABLE RCA13_DOCS (
  DOC_ID          NUMBER DEFAULT
to_number(sys_guid(), 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX') NOT NULL ENABLE,
  COLLECTION_ID  VARCHAR2(40 BYTE),
  FILENAME       VARCHAR2(1000 BYTE) NOT NULL ENABLE,
  FILE_MIMETYPE  VARCHAR2(512 BYTE),
  FILE_CHARSET   VARCHAR2(512 BYTE),
  FILE_BLOB      BLOB NOT NULL ENABLE,
  FILE_COMMENTS  VARCHAR2(4000 BYTE),
  TAGS           VARCHAR2(4000 BYTE),
  ATTR1         VARCHAR2(200 BYTE),
  UPLOADED_BY    VARCHAR2(200 BYTE) DEFAULT USER,
  UPLOADED_ON    TIMESTAMP (6) DEFAULT systimestamp,
  SR_BUG_NUM     VARCHAR2(20 BYTE),
  CONSTRAINT RCA13_DOCS_PK PRIMARY KEY (DOC_ID),
```

```

        CONSTRAINT RCA13_DOCS_UK1 UNIQUE (FILENAME)
    );

```

- **DDL for the auditcheck_result table**

```

CREATE TABLE auditcheck_result (
    COLLECTION_DATE          TIMESTAMP NOT NULL ENABLE,
    CHECK_NAME               VARCHAR2(256),
    PARAM_NAME               VARCHAR2(256),
    STATUS                   VARCHAR2(256),
    STATUS_MESSAGE           VARCHAR2(256),
    ACTUAL_VALUE             VARCHAR2(256),
    RECOMMENDED_VALUE        VARCHAR2(256),
    COMPARISON_OPERATOR      VARCHAR2(256),
    HOSTNAME                 VARCHAR2(256),
    INSTANCE_NAME            VARCHAR2(256),
    CHECK_TYPE               VARCHAR2(256),
    DB_PLATFORM              VARCHAR2(256),
    OS_DISTRO                VARCHAR2(256),
    OS_KERNEL                VARCHAR2(256),
    OS_VERSION               NUMBER,
    DB_VERSION               VARCHAR2(256),
    CLUSTER_NAME             VARCHAR2(256),
    DB_NAME                  VARCHAR2(256),
    ERROR_TEXT               VARCHAR2(256),
    CHECK_ID                 VARCHAR2(40),
    NEEDS_RUNNING            VARCHAR2(100),
    MODULES                  VARCHAR2(4000),
    DATABASE_ROLE            VARCHAR2(100),
    CLUSTERWARE_VERSION      VARCHAR2(100),
    GLOBAL_NAME              VARCHAR2(256),
    UPLOAD_COLLECTION_NAME    VARCHAR2(256) NOT NULL ENABLE,
    AUDITCHECK_RESULT_ID     VARCHAR2(256) DEFAULT sys_guid() NOT NULL
    ENABLE,
    COLLECTION_ID            VARCHAR2(40),
    TARGET_TYPE              VARCHAR2(128),
    TARGET_VALUE             VARCHAR2(256),
    CONSTRAINT "AUDITCHECK_RESULT_PK" PRIMARY KEY ("AUDITCHECK_RESULT_ID")
);

```

- **DDL for the auditcheck_patch_result table**

```

CREATE TABLE auditcheck_patch_result (
    COLLECTION_DATE          TIMESTAMP(6) NOT NULL,
    HOSTNAME                 VARCHAR2(256),
    ORACLE_HOME_TYPE         VARCHAR2(256),
    ORACLE_HOME_PATH         VARCHAR2(256),
    ORACLE_HOME_VERSION      VARCHAR2(256),
    PATCH_NUMBER             NUMBER,
    CLUSTER_NAME             VARCHAR2(256),
    DESCRIPTION              VARCHAR2(256),
    PATCH_TYPE               VARCHAR2(128),
    APPLIED                  NUMBER,
    UPLOAD_COLLECTION_NAME    VARCHAR2(256),
    RECOMMENDED              NUMBER
);

```

Related Topics

- [Uploading Collections Automatically](#)
Configure Oracle ORAchk and Oracle EXAchk to upload check results automatically to the Oracle Health Check Collections Manager database.

2.11 Troubleshooting Oracle ORAchk and Oracle EXAchk

To troubleshoot and fix Oracle ORAchk and Oracle EXAchk issues, follow the steps explained in this section.

- [How to Troubleshoot Oracle ORAchk and Oracle EXAchk Issues](#)
To troubleshoot Oracle ORAchk and Oracle EXAchk issues, follow the steps explained in this section.
- [How to Capture Debug Output](#)
Follow these steps to capture debug information.
- [Remote Login Problems](#)
If Oracle ORAchk and Oracle EXAchk tools have problem locating and running SSH or SCP, then the tools cannot run any remote checks.
- [Permission Problems](#)
You must have sufficient directory permissions to run Oracle ORAchk and Oracle EXAchk.
- [Slow Performance, Skipped Checks and Timeouts](#)
Follow these steps to fix slow performance and other issues.

Related Topics

- *Oracle ORAchk and EXAchk User's Guide*

2.11.1 How to Troubleshoot Oracle ORAchk and Oracle EXAchk Issues

To troubleshoot Oracle ORAchk and Oracle EXAchk issues, follow the steps explained in this section.

To troubleshoot Oracle ORAchk and Oracle EXAchk:

1. Ensure that you are using the correct tool.
Use Oracle EXAchk for Oracle Engineered Systems except for Oracle Database Appliance. For all other systems, use Oracle ORAchk.
2. Ensure that you are using the latest versions of Oracle ORAchk and Oracle EXAchk.
 - a. Check the version using the `-v` option.

```
$ ./orachk -v
```

```
$ ./exachk -v
```
 - b. Compare your version with the latest version available here:
 - For Oracle ORAchk, refer to My Oracle Support Note 1268927.2.
 - For Oracle EXAchk, refer to My Oracle Support Note 1070954.1.
3. Check the **FAQ** for similar problems in My Oracle Support Note 1070954.1.
4. Review the files within the `log` directory.
 - Check the applicable `error.log` files for relevant errors.
The `error.log` files contain `stderr` output captured during the run.

- `output_dir/log/orachk_error.log`
- `output_dir/log/exachk_error.log`
- Check the applicable log for other relevant information.
 - `output_dir/log/orachk.log`
 - `output_dir/log/exachk.log`
- 5. Review My Oracle Support Notes for similar problems.
- 6. For Oracle ORAchk issues, check ORAchk (MOSC) in My Oracle Support Community (MOSC).
- 7. If necessary, capture the debug output, and then log an SR and attach the resulting `zip` file.

Related Topics

- <https://support.oracle.com/rs?type=doc&id=1268927.2>
- <https://support.oracle.com/rs?type=doc&id=1070954.1>

2.11.2 How to Capture Debug Output

Follow these steps to capture debug information.

To capture debug output:

1. Reproduce the problem with fewest runs before enabling debug.

Debug captures a lot and the resulting `zip` file can be large so try to narrow down the amount of run necessary to reproduce the problem.

Use command-line options to limit the scope of checks.

2. Enable debug.

If you are running the tool in on-demand mode, then use the `-debug` option:

```
$ ./orachk -debug
```

```
$ ./exachk -debug
```

When you enable debug, Oracle ORAchk and Oracle EXAchk create a new debug log file in:

- `output_dir/log/orachk_debug_date_stamp_time_stamp.log`
- `output_dir/log/exachk_debug_date_stamp_time_stamp.log`

The `output_dir` directory retains various other temporary files used during health checks.

If you run health checks using the daemon, then restart the daemon with the `-d start -debug` option.

Running this command generates both debug for daemon and include debug in all client runs:

```
$ ./orachk -d start -debug
```

```
$ ./exachk -d start -debug
```

When debug is run with the daemon, Oracle ORAchk and Oracle EXAchk create a daemon debug log file in the directory in which the daemon was started:

```
orachk_daemon_debug.log
```

```
exachk_daemon_debug.log
```

3. Collect the resulting output zip file and the daemon debug log file, if applicable.

2.11.3 Remote Login Problems

If Oracle ORAchk and Oracle EXAchk tools have problem locating and running SSH or SCP, then the tools cannot run any remote checks.

Also, the `root` privileged commands do not work if:

- Passwordless remote `root` login is not permitted over SSH
- Expect utility is not able to pass the `root` password

1. Verify that the SSH and SCP commands can be found.

- The SSH commands return the error, `-bash: /usr/bin/ssh -q: No such file or directory`, if SSH is not located where expected.

Set the `RAT_SHELL` environment variable pointing to the location of SSH:

```
$ export RAT_SHELL=path to ssh
```

- The SCP commands return the error, `/usr/bin/scp -q: No such file or directory`, if SCP is not located where expected.

Set the `RAT_COPY` environment variable pointing to the location of SCP:

```
$ export RAT_COPY=path to scp
```

2. Verify that the user you are running as, can run the following command manually from where you are running Oracle ORAchk and Oracle EXAchk to whichever remote node is failing.

```
$ ssh root@remotehostname "id"
root@remotehostname's password:
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),
10(wheel)
```

- If you face any problems running the command, then contact the systems administrators to correct temporarily for running the tool.
- Oracle ORAchk and Oracle EXAchk search for the prompts or traps in remote user profiles. If you have prompts in remote profiles, then comment them out at least temporarily and test run again.
- If you can configure passwordless remote `root` login, then edit the `/etc/ssh/sshd_config` file as follows:

```
n to yes
```

Now, run the following command as `root` on all nodes of the cluster:

```
hd restart
```

3. Enable Expect debugging.

- Oracle ORAchk uses the Expect utility when available to answer password prompts to connect to remote nodes for password validation. Also, to run `root` collections without logging the actual connection process by default.
- Set environment variables to help debug remote target connection issues.
 - **RAT_EXPECT_DEBUG**: If this variable is set to `-d`, then the Expect command tracing is activated. The trace information is written to the standard output.

For example:

```
export RAT_EXPECT_DEBUG=-d
```

- **RAT_EXPECT_STRACE_DEBUG**: If this variable is set to `strace`, `strace` calls the Expect command. The trace information is written to the standard output.

For example:

```
export RAT_EXPECT_STRACE_DEBUG=strace
```

- By varying the combinations of these two variables, you can get three levels of Expect connection trace information.



Note:

Set the `RAT_EXPECT_DEBUG` and `RAT_EXPECT_STRACE_DEBUG` variables only at the direction of Oracle support or development. The `RAT_EXPECT_DEBUG` and `RAT_EXPECT_STRACE_DEBUG` variables are used with other variables and user interface options to restrict the amount of data collected during the tracing. The `script` command is used to capture standard output.

As a temporary workaround while you resolve remote problems, run reports local on each node then merge them together later.

On each node, run:

```
./orachk -local
```

```
./exachk -local
```

Then merge the collections to obtain a single report:

```
./orachk -merge zipfile 1 zipfile 2 > zipfile 3 > zipfile ...
```

```
./exachk -merge zipfile 1 zipfile 2 > zipfile 3 > zipfile ...
```

2.11.4 Permission Problems

You must have sufficient directory permissions to run Oracle ORAchk and Oracle EXAchk.

1. Verify that the permissions on the tools scripts `orachk` and `exachk` are set to 755 (`-rwxr-xr-x`).

If the permissions are not set, then set the permissions as follows:

```
$ chmod 755 orachk
```

```
$ chmod 755 exachk
```

2. If you install Oracle ORAchk and Oracle EXAchk as `root` and run the tools as a different user, then you may not have the necessary directory permissions.

```
[root@randomdb01 exachk]# ls -la
total 14072
drwxr-xr-x  3 root root   4096 Jun  7 08:25 .
drwxrwxrwt 12 root root   4096 Jun  7 09:27 ..
drwxrwxr-x  2 root root   4096 May 24 16:50 .cgrep
-rw-rw-r--  1 root root 9099005 May 24 16:50 collections.dat
-rwxr-xr-x  1 root root  807865 May 24 16:50 exachk
-rw-r--r--  1 root root 1646483 Jun  7 08:24 exachk.zip
-rw-r--r--  1 root root   2591 May 24 16:50 readme.txt
-rw-rw-r--  1 root root 2799973 May 24 16:50 rules.dat
-rw-r--r--  1 root root    297 May 24 16:50 UserGuide.txt
```

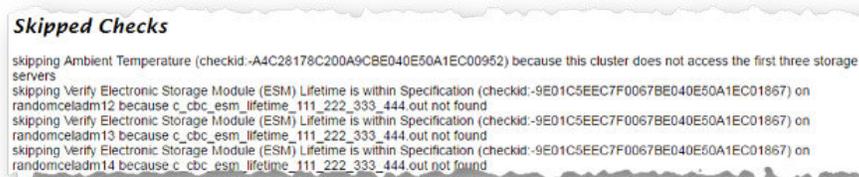
In which case, you must run as `root` or unzip again as the Oracle software install user.

2.11.5 Slow Performance, Skipped Checks and Timeouts

Follow these steps to fix slow performance and other issues.

When Oracle ORAchk and Oracle EXAchk run commands, a child process is spawned to run the command and a watchdog daemon monitors the child process. If the child process is slow or hung, then the watchdog kills the child process and the check is registered as skipped:

Figure 2-27 Skipped Checks



```
Skipped Checks
skipping Ambient Temperature (checkid:-A4C28178C200A9CBE040E50A1EC00952) because this cluster does not access the first three storage servers
skipping Verify Electronic Storage Module (ESM) Lifetime is within Specification (checkid:-9E01C5EEC7F0067BE040E50A1EC01867) on randomceladm12 because c_cbc_esm_lifetime_111_222_333_444.out not found
skipping Verify Electronic Storage Module (ESM) Lifetime is within Specification (checkid:-9E01C5EEC7F0067BE040E50A1EC01867) on randomceladm13 because c_cbc_esm_lifetime_111_222_333_444.out not found
skipping Verify Electronic Storage Module (ESM) Lifetime is within Specification (checkid:-9E01C5EEC7F0067BE040E50A1EC01867) on randomceladm14 because c_cbc_esm_lifetime_111_222_333_444.out not found
```

The `watchdog.log` file also contains entries similar to **killing stuck command**.

Depending on the cause of the problem, you may not see skipped checks.

1. Determine if there is a pattern to what is causing the problem.
 - EBS checks, for example, depend on the amount of data present and may take longer than the default timeout.
 - Remote checks may timeout and be killed and skipped, if there are prompts in the remote profile. Oracle ORAchk and Oracle EXAchk search for prompts or traps in the remote user profiles. If you have prompts in remote profiles, then comment them out at least temporarily and test run again.
2. Increase the default timeout.
 - Override the default timeout by setting the environment variables.

Table 2-4 Timeout Controlling

Timeout Controlling	Default Value (seconds)	Environment Variable
Checks not run by root (most).	90	RAT_TIMEOUT
Collection of all root checks.	300	RAT_ROOT_TIMEOUT
SSH login DNS handshake.	1	RAT_PASSWORDCHECK_TIMEOUT

- The default timeouts are designed to be lengthy enough for most cases. If the timeout is not long enough, then it is possible you are experiencing a system performance problem. Many timeouts can be indicative of a non-Oracle ORAchk and Oracle EXAchk problem in the environment.
3. If it is not acceptable to increase the timeout to the point where nothing fails, then try excluding problematic checks running separately with a large enough timeout and then merging the reports back together.
 4. If the problem does not appear to be down to slow or skipped checks but you have a large cluster, then try increasing the number of slave processes user for parallel database run.
 - Database collections are run in parallel. The default number of slave processes used for parallel database run is calculated automatically. Change the default number using the options: `-dbparallel slave processes`, or `-dbparallelmax`

 **Note:**

The higher the parallelism the more resources are consumed. However, the elapsed time is reduced.

Raise or lower the number of parallel slaves beyond the default value.

After the entire system is brought up after maintenance, but before the users are permitted on the system, use a higher number of parallel slaves to finish a run as quickly as possible.

On a busy production system, use a number less than the default value yet more than running in serial mode to get a run more quickly with less impact on the running system.

Turn off the parallel database run using the `-dbserial` option.

3

Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

Oracle Cluster Health Advisor provides system and database administrators with early warning of pending performance issues, and root causes and corrective actions for Oracle RAC databases and cluster nodes.

Use Oracle Cluster Health Advisor to increase availability and performance management.

Oracle Cluster Health Advisor estimates an expected value of an observed input based on the default model, which is a trained calibrated model based on a normal operational period of the target system. Oracle Cluster Health Advisor then performs anomaly detection for each input based on the difference between observed and expected values. If sufficient inputs associated with a specific problem are abnormal, then Oracle Cluster Health Advisor raises a warning and generates an immediate targeted diagnosis and corrective action.

Oracle Cluster Health Advisor stores the analysis results, along with diagnosis information, corrective action, and metric evidence for later triage, in the Grid Infrastructure Management Repository (GIMR). Oracle Cluster Health Advisor also sends warning messages to Enterprise Manager Cloud Control using the Oracle Clusterware event notification protocol.

- [Oracle Cluster Health Advisor Architecture](#)
Oracle Cluster Health Advisor runs as a highly available cluster resource, `ochad`, on each node in the cluster.
- [Monitoring the Oracle Real Application Clusters \(Oracle RAC\) Environment with Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor is automatically provisioned on each node by default when Oracle Grid Infrastructure is installed for Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database.
- [Using Cluster Health Advisor for Health Diagnosis](#)
Oracle Cluster Health Advisor raises and clears problems autonomously and stores the history in the Grid Infrastructure Management Repository (GIMR).
- [Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment](#)
As shipped with default node and database models, Oracle Cluster Health Advisor is designed not to generate false warning notifications.
- [Viewing the Details for an Oracle Cluster Health Advisor Model](#)
Use the `chactl query model` command to view the model details.
- [Managing the Oracle Cluster Health Advisor Repository](#)
Oracle Cluster Health Advisor repository stores the historical records of cluster host problems, database problems, and associated metric evidence, along with models.
- [Viewing the Status of Cluster Health Advisor](#)
SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.

Related Topics

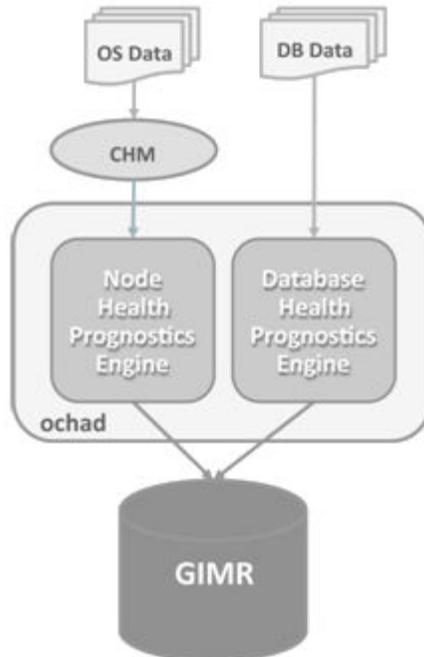
- [Introduction to Oracle Cluster Health Advisor](#)
Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.

3.1 Oracle Cluster Health Advisor Architecture

Oracle Cluster Health Advisor runs as a highly available cluster resource, `ochad`, on each node in the cluster.

Each Oracle Cluster Health Advisor daemon (`ochad`) monitors the operating system on the cluster node and optionally, each Oracle Real Application Clusters (Oracle RAC) database instance on the node.

Figure 3-1 Oracle Cluster Health Advisor Architecture



The `ochad` daemon receives operating system metric data from the Cluster Health Monitor and gets Oracle RAC database instance metrics from a memory-mapped file. The daemon does not require a connection to each database instance. This data, along with the selected model, is used in the Health Prognostics Engine of Oracle Cluster Health Advisor for both the node and each monitored database instance in order to analyze their health multiple times a minute.

3.2 Monitoring the Oracle Real Application Clusters (Oracle RAC) Environment with Oracle Cluster Health Advisor

Oracle Cluster Health Advisor is automatically provisioned on each node by default when Oracle Grid Infrastructure is installed for Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database.

Oracle Cluster Health Advisor does not require any additional configuration. The credentials of OCHAD daemon user in the Grid Infrastructure Management Repository (GIMR), are securely and randomly generated and stored in the Oracle Grid Infrastructure Credential Store.

When Oracle Cluster Health Advisor detects an Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database instance as running, Oracle Cluster Health Advisor autonomously starts monitoring the cluster nodes. Use CHACTL while logged in as the Grid user to turn on monitoring of the database.

To monitor the Oracle Real Application Clusters (Oracle RAC) environment:

1. To monitor a database, run the following command:

```
$ chactl monitor database -db db_unique_name
```

Oracle Cluster Health Advisor monitors all instances of the Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node database using the default model. Oracle Cluster Health Advisor cannot monitor single-instance Oracle databases, even if the single-instance Oracle databases share the same cluster as Oracle Real Application Clusters (Oracle RAC) databases.

Oracle Cluster Health Advisor preserves database monitoring status across cluster restarts as Oracle Cluster Health Advisor stores the status information in the GIMR. Each database instance is monitored independently both across Oracle Real Application Clusters (Oracle RAC) database nodes and when more than one database run on a single node.

2. To stop monitoring a database, run the following command:

```
$ chactl unmonitor database -db db_unique_name
```

Oracle Cluster Health Advisor stops monitoring all instances of the specified database. However, Oracle Cluster Health Advisor does not delete any data or problems until it is aged out beyond the retention period.

3. To check monitoring status of all cluster nodes and databases, run the following command:

```
$ chactl status
```

Use the `-verbose` option to see more details, such as the models used for the nodes and each database.

3.3 Using Cluster Health Advisor for Health Diagnosis

Oracle Cluster Health Advisor raises and clears problems autonomously and stores the history in the Grid Infrastructure Management Repository (GIMR).

The Oracle Grid Infrastructure user can query the stored information using CHACTL.

To query the diagnostic data:

1. To query currently open problems, run the following command:

```
chactl query diagnosis -db db_unique_name -start time -end time
```

In the syntax example, *db_unique_name* is the name of your database instance. You also specify the start time and end time for which you want to retrieve data. Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

2. Use the `-htmlfile file_name` option to save the output in HTML format.

Example 3-1 Cluster Health Advisor Output Examples in Text and HTML Format

This example shows the default text output for the `chactl query diagnosis` command for a database named *oltpacdb*.

```
$ chactl query diagnosis -db oltpacdb -start "2016-02-01 02:52:50" -end "2016-02-01 03:19:15"
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_1) [detected]
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_2) [detected]
2016-02-01 02:52:15.0 Database oltpacdb DB CPU Utilization (oltpacdb_2) [detected]
2016-02-01 02:52:50.0 Database oltpacdb DB CPU Utilization (oltpacdb_1) [detected]
2016-02-01 02:59:35.0 Database oltpacdb DB Log File Switch (oltpacdb_1) [detected]
2016-02-01 02:59:45.0 Database oltpacdb DB Log File Switch (oltpacdb_2) [detected]
```

Problem: DB Control File IO Performance

Description: CHA has detected that reads or writes to the control files are slower than expected.

Cause: The Cluster Health Advisor (CHA) detected that reads or writes to the control files were slow because of an increase in disk IO.

The slow control file reads and writes may have an impact on checkpoint and Log Writer (LGWR) performance.

Action: Separate the control files from other database files and move them to faster disks or Solid State Devices.

Problem: DB CPU Utilization

Description: CHA detected larger than expected CPU utilization for this database.

Cause: The Cluster Health Advisor (CHA) detected an increase in database CPU utilization

because of an increase in the database workload.

Action: Identify the CPU intensive queries by using the Automatic Diagnostic and Defect Manager (ADDM) and

follow the recommendations given there. Limit the number of CPU intensive queries or relocate sessions to less busy machines. Add CPUs if the CPU capacity is insufficient to support

the load without a performance degradation or effects on other databases.

Problem: DB Log File Switch

Description: CHA detected that database sessions are waiting longer than expected for log switch completions.

Cause: The Cluster Health Advisor (CHA) detected high contention during log switches because the redo log files were small and the redo logs switched frequently.

Action: Increase the size of the redo logs.

The timestamp displays date and time when the problem was detected on a specific host or database.

Note:

The same problem can occur on different hosts and at different times, yet the diagnosis shows complete details of the problem and its potential impact. Each problem also shows targeted corrective or preventive actions.

Here is an example of what the output looks like in the HTML format.

```
$ chactl query diagnosis -start "2016-07-03 20:50:00" -end "2016-07-04 03:50:00" -
htmlfile ~/chaprob.html
```

Figure 3-2 Cluster Health Advisor Diagnosis HTML Output

Timestamp	Target Information	Event Name	Detected/Cleared
2016-07-03 01:49:30.0	Host rwsbi07	Host CPU Utilization	detected
2016-07-03 01:49:50.0	Host rwsbi06	Host CPU Utilization	detected
2016-07-03 05:54:55.0	Host rwsbi06	Host Memory Consumption	detected
2016-07-04 03:40:00.0	Host rwsbi07	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host Memory Consumption	cleared

Problem	Description	Cause	Action
Host CPU Utilization	CHA detected larger than expected CPU utilization on this node. The available CPU resource may not be sufficient to support application failover or relocation of databases to this node.	The Cluster Health Advisor (CHA) detected an unexpected increase in CPU utilization by databases or applications on this node.	Identify CPU intensive processes and databases by reviewing Cluster Health Monitoring (CHM) data. Relocate databases to less busy machines, or limit the number of connections to databases on this node. Add nodes if more resources are required.
Host Memory Consumption	CHA detected that more memory than expected is consumed on this server. The memory is not allocated by sessions of this database.	The Cluster Health Advisor (CHA) detected an increase in memory consumption by other databases or by applications not connected to a database on this node.	Identify the top memory consumers by using the Cluster Health Monitor (CHM).

Related Topics

- [chactl query diagnosis](#)
Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.

3.4 Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment

As shipped with default node and database models, Oracle Cluster Health Advisor is designed not to generate false warning notifications.

You can increase the sensitivity and accuracy of the Oracle Cluster Health Advisor models for a specific workload using the `chactl calibrate` command.

Oracle recommends that a minimum of 6 hours of data be available and that both the cluster and databases use the same time range for calibration.

The `chactl calibrate` command analyzes a user-specified time interval that includes all workload phases operating normally. This data is collected while Oracle Cluster Health Advisor is monitoring the cluster and all the databases for which you want to calibrate.

1. To check if sufficient data is available, run the `query calibration` command.

If 720 or more records are available, then Oracle Cluster Health Advisor successfully performs the calibration. The calibration function may not consider some data records to be normally occurring for the workload profile being used. In this case, filter the data by using the `KPISET` parameters in both the `query calibration` command and the `calibrate` command.

For example:

```
$ chactl query calibration -db oltpacdb -timeranges
'start=2016-07-26 01:00:00,end=2016-07-26 02:00:00,start=2016-07-26
03:00:00,end=2016-07-26 04:00:00'
-kpiset 'name=CPUPERCENT min=20 max=40, name=IOTHROUGHPUT min=500 max=9000' -
interval 2
```

2. Start the calibration and store the model under a user-specified name for the specified date and time range.

For example:

```
$ chactl calibrate cluster -model weekday -timeranges 'start=2016-07-03
20:50:00,end=2016-07-04 15:00:00'
```

After completing the calibration, Oracle Cluster Health Advisor automatically stores the new model in GIMR.

3. Use the new model to monitor the cluster as follows:

For example:

```
$ chactl monitor cluster -model weekday
```

Example 3-2 Output for the `chactl query calibrate` command

```
Database name : oltpacdb
Start time : 2016-07-26 01:03:10
End time : 2016-07-26 01:57:25
Total Samples : 120
Percentage of filtered data : 8.32%
The number of data samples may not be sufficient for calibration.
```

```
1) Disk read (ASM) (Mbyte/sec)
```

MEAN	MEDIAN	STDDEV	MIN	MAX
4.96	0.20	8.98	0.06	25.68
<25	<50	<75	<100	>=100
97.50%	2.50%	0.00%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
27.73	9.72	31.75	4.16	109.39
<50	<100	<150	<200	>=200
73.33%	22.50%	4.17%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
2407.50	1500.00	1978.55	700.00	7800.00
<5000	<10000	<15000	<20000	>=20000
83.33%	16.67%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
21.99	21.75	1.36	20.00	26.80
<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX			
267.39	264.87	32.05	205.80	484.57			
<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000	>=70000000
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Database name : oltpacdb

Start time : 2016-07-26 03:00:00

End time : 2016-07-26 03:53:30

Total Samples : 342

Percentage of filtered data : 23.72%

The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
12.18	0.28	16.07	0.05	60.98
<25	<50	<75	<100	>=100
64.33%	34.50%	1.17%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
57.57	51.14	34.12	16.10	135.29
<50	<100	<150	<200	>=200

```
49.12%    38.30%    12.57%    0.00%    0.00%
```

```
3) Disk throughput (ASM) (IO/sec)
```

```
MEAN      MEDIAN    STDDEV    MIN      MAX
5048.83   4300.00   1730.17   2700.00  9000.00
```

```
<5000    <10000    <15000    <20000    >=20000
63.74%   36.26%   0.00%     0.00%     0.00%
```

```
4) CPU utilization (total) (%)
```

```
MEAN      MEDIAN    STDDEV    MIN      MAX
23.10     22.80     1.88      20.00    31.40
```

```
<20      <40      <60      <80      >=80
0.00%    100.00%  0.00%    0.00%    0.00%
```

```
5) Database time per user call (usec/call)
```

```
MEAN      MEDIAN    STDDEV    MIN      MAX
744.39    256.47    2892.71   211.45   45438.35
```

```
<10000000 <20000000 <30000000 <40000000 <50000000 <60000000 <70000000
>=70000000
100.00%   0.00%     0.00%     0.00%     0.00%     0.00%     0.00%
```

Related Topics

- [chactl calibrate](#)
Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- [chactl query calibration](#)
Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- [chactl Command Reference](#)
The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

3.5 Viewing the Details for an Oracle Cluster Health Advisor Model

Use the `chactl query model` command to view the model details.

1. You can review the details of an Oracle Cluster Health Advisor model at any time using the `chactl query model` command.

For example:

```
$ chactl query model -name weekday
Model: weekday
Target Type: CLUSTERWARE
Version: OS12.2_V14_0.9.8
OS Calibrated on: Linux amd64
Calibration Target Name: MYCLUSTER
Calibration Date: 2016-07-05 01:13:49
```

```
Calibration Time Ranges: start=2016-07-03 20:50:00,end=2016-07-04 15:00:00  
Calibration KPIs: not specified
```

You can also rename, import, export, and delete the models.

3.6 Managing the Oracle Cluster Health Advisor Repository

Oracle Cluster Health Advisor repository stores the historical records of cluster host problems, database problems, and associated metric evidence, along with models.

The Oracle Cluster Health Advisor repository is used to diagnose and triage periodic problems. By default, the repository is sized to retain data for 16 targets (nodes and database instances) for 72 hours. If the number of targets increase, then the retention time is automatically decreased. Oracle Cluster Health Advisor generates warning messages when the retention time goes below 72 hours, and stops monitoring and generates a critical alert when the retention time goes below 24 hours.

Use CHACTL commands to manage the repository and set the maximum retention time.

1. To retrieve the repository details, use the following command:

```
$ chactl query repository
```

For example, running the command mentioned earlier shows the following output:

```
specified max retention time(hrs) : 72  
available retention time(hrs)      : 212  
available number of entities       : 2  
allocated number of entities       : 0  
total repository size(gb)          : 2.00  
allocated repository size(gb)      : 0.07
```

2. To set the maximum retention time in hours, based on the current number of targets being monitored, use the following command:

```
$ chactl set maxretention -time number_of_hours
```

For example:

```
$ chactl set maxretention -time 80  
max retention successfully set to 80 hours
```

Note:

The `maxretention` setting limits the oldest data retained in the repository, but is not guaranteed to be maintained if the number of monitored targets increase. In this case, if the combination of monitored targets and number of hours are not sufficient, then increase the size of the Oracle Cluster Health Advisor repository.

3. To increase the size of the Oracle Cluster Health Advisor repository, use the `chactl resize repository` command.

For example, to resize the repository to support 32 targets using the currently set maximum retention time, you would use the following command:

```
$ chactl resize repository -entities 32  
repository successfully resized for 32 targets
```

3.7 Viewing the Status of Cluster Health Advisor

SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.

Use SRVCTL commands to check the status and configuration of Oracle Cluster Health Advisor service on any active hub or leaf nodes of the Oracle RAC cluster.



Note:

A target is monitored only if it is running and the Oracle Cluster Health Advisor service is also running on the host node where the target exists.

1. To check the status of Oracle Cluster Health Advisor service on all nodes in the Oracle RAC cluster:

```
srvctl status cha [-help]
```

For example:

```
# srvctl status cha  
Cluster Health Advisor is running on nodes racNode1, racNode2.  
Cluster Health Advisor is not running on nodes racNode3, racNode4.
```

2. To check if Oracle Cluster Health Advisor service is enabled or disabled on all nodes in the Oracle RAC cluster:

```
srvctl config cha [-help]
```

For example:

```
# srvctl config cha  
Cluster Health Advisor is enabled on nodes racNode1, racNode2.  
Cluster Health Advisor is not enabled on nodes racNode3, racNode4.
```

Part II

Automatically Monitoring the Cluster

You can use components of Autonomous Health Framework to monitor your cluster on a regular basis.

- [Collecting Operating System Resources Metrics](#)
Use Cluster Health Monitor to collect diagnostic data to analyze performance degradation or failures of critical operating system resources.
- [Monitoring System Metrics for Cluster Nodes](#)
This chapter explains the methods to monitor Oracle Clusterware.

4

Collecting Operating System Resources Metrics

Use Cluster Health Monitor to collect diagnostic data to analyze performance degradation or failures of critical operating system resources.

- [Understanding Cluster Health Monitor Services](#)
Cluster Health Monitor uses system monitor (`osysmond`) and cluster logger (`ologgerd`) services to collect diagnostic data.
- [Collecting Cluster Health Monitor Data](#)
Collect Cluster Health Monitor data from any node in the cluster.
- [Using Cluster Health Monitor from Enterprise Manager Cloud Control](#)
Histograms presented in real-time and historical modes enable you to understand precisely what was happening at the time of degradation or failure.

Related Topics

- [Introduction to Cluster Health Monitor](#)
Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.

4.1 Understanding Cluster Health Monitor Services

Cluster Health Monitor uses system monitor (`osysmond`) and cluster logger (`ologgerd`) services to collect diagnostic data.

About the System Monitor Service

The system monitor service (`osysmond`) is a real-time monitoring and operating system metric collection service that runs on each cluster node. The system monitor service is managed as a High Availability Services (HAS) resource. The system monitor service forwards the collected metrics to the cluster logger service, `ologgerd`. The cluster logger service stores the data in the Oracle Grid Infrastructure Management Repository database.

About the Cluster Logger Service

The cluster logger service (`ologgerd`) is responsible for preserving the data collected by the system monitor service (`osysmond`) in the Oracle Grid Infrastructure Management Repository database. In a cluster, there is one cluster logger service (`ologgerd`) per 32 nodes. More logger services are spawned for every additional 32 nodes. The additional nodes can be a sum of Hub and Leaf Nodes. Oracle Clusterware relocates and starts the service on a different node, if:

- The logger service fails and is not able to come up after a fixed number of retries
- The *node* where the cluster logger service is running, is down

4.2 Collecting Cluster Health Monitor Data

Collect Cluster Health Monitor data from any node in the cluster.

Oracle recommends that you run the `tfactl diagcollect` command to collect diagnostic data when an Oracle Clusterware error occurs.

4.3 Using Cluster Health Monitor from Enterprise Manager Cloud Control

Histograms presented in real-time and historical modes enable you to understand precisely what was happening at the time of degradation or failure.

The metric data from Cluster Health Monitor is available in graphical display within Enterprise Manager Cloud Control. Complete cluster views of this data are accessible from the cluster target page. Selecting the **Cluster Health Monitoring** menu item from the **Cluster** menu presents a log-in screen prompting for the Cluster Health Monitor credentials. There is a fixed EMUSER and the password is user-specified. Once the credentials are saved, you then can view Cluster Health Monitor data for the last day in overview format for the entire cluster. Metric categories are CPU, Memory, and Network.

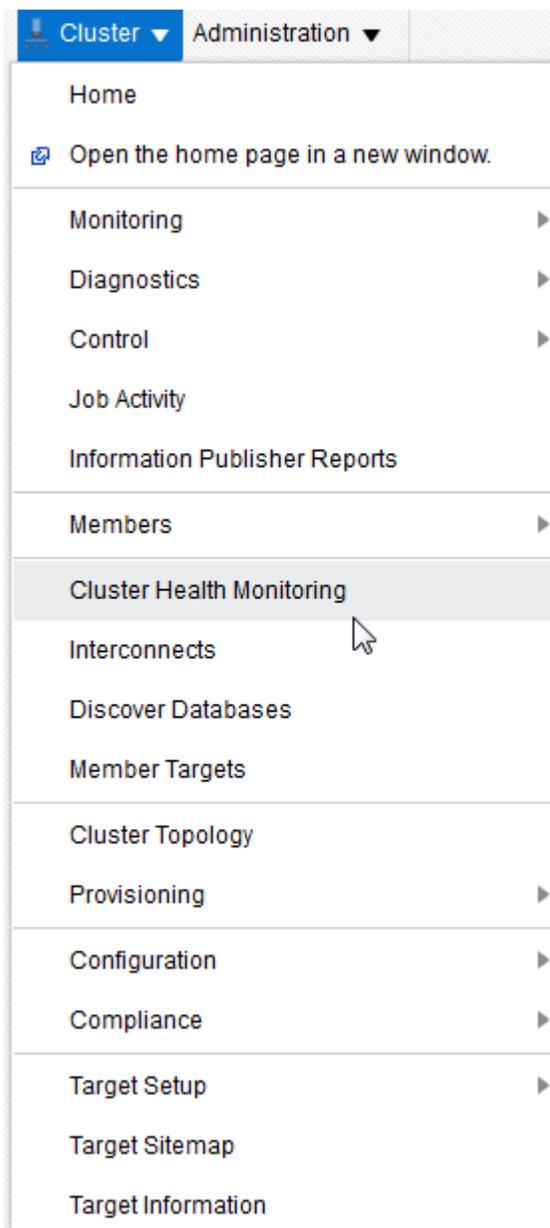
Each category is able to be separately display in greater detail showing more metrics. For example, selecting CPU results in cluster graphs detailing CPU System Usage, CPU User Usage, and CPU Queue Length. From any cluster view, you can select individual node views to more closely examine performance of a single server. As in the case of CPU, the performance of each core is displayed. Move your cursor along the graph to see a tool-tip displaying the numerical values and time stamp of that point.

Besides examining the performance of the current day, you can also review historical data. The amount of historical data is governed by the retention time configured in the Cluster Health Monitor repository in the Grid Infrastructure Management Repository and defaults to 72 hours. This view is selectable at any time by using the **View Mode** drop-down menu and selecting **Historical**. A previous date can then be entered or selected from a pop-up calendar that has dates where data is available bolded. Selecting **Show Chart** then displays the associated metrics graphs.

To view Cluster Health Monitor data:

1. Log in to Enterprise Manager Cloud Control.
2. Select the Cluster Target you want to view.
3. From the **Cluster** drop-down list, select the **Cluster Health Monitoring** option.

Figure 4-1 EMCC - Cluster Health Monitoring

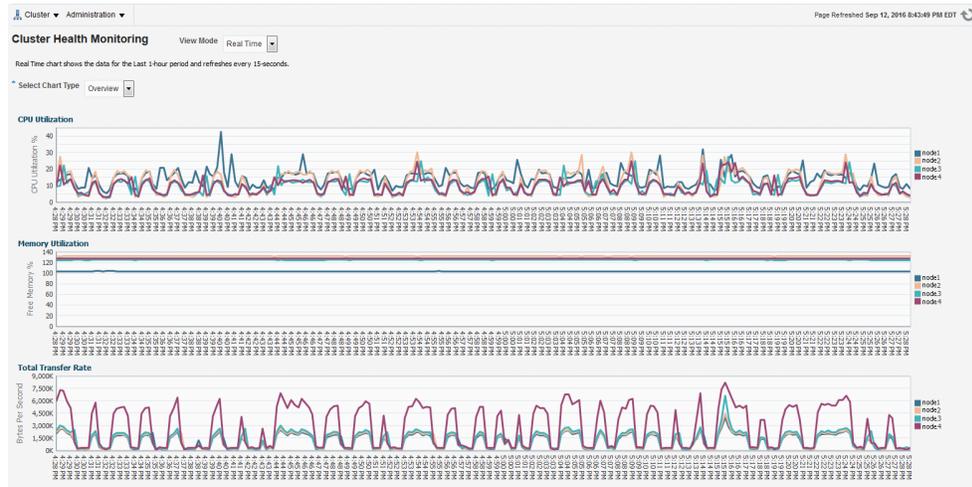


4. Enter Cluster Health Monitor login credentials.
5. From the **View Mode** drop-down list, select the **Real Time** option to view the current data.

By default, EMCC displays the **Overview** of resource utilization. You can filter by **CPU**, **Memory**, and **Network** by selecting an appropriate option from the **Select Chart Type** drop-down list.

While viewing CPU and Network metric graphs, click a node name on the legend to view more details.

Figure 4-2 Cluster Health Monitoring - Real Time Data

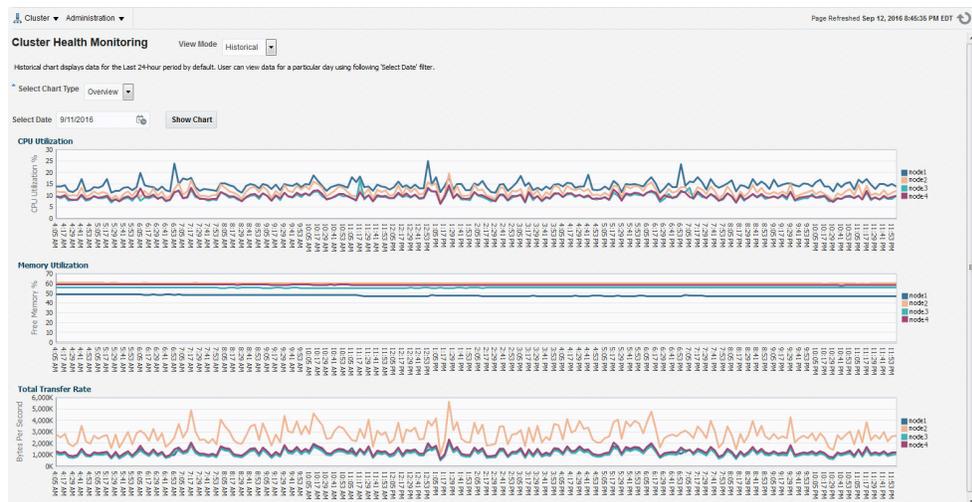


6. From the **View Mode** drop-down list, select the **Historical** option to view data for the last 24 hours.
 - a. To filter historical data by date, select a day on the **Select Date** calendar control and then click **Show Chart**.

By default, EMCC displays the **Overview** of resource utilization. You can filter by **CPU**, **Memory**, and **Network** by selecting an appropriate option from the **Select Chart Type** drop-down list.

While viewing CPU and Network metric graphs, click a node name on the legend to view more details.

Figure 4-3 Cluster Health Monitoring - Historical Data



5

Monitoring System Metrics for Cluster Nodes

This chapter explains the methods to monitor Oracle Clusterware.

Oracle recommends that you use Oracle Enterprise Manager to monitor everyday operations of Oracle Clusterware.

Cluster Health Monitor monitors the complete technology stack, including the operating system, ensuring smooth cluster operations. Both the components are enabled, by default, for any Oracle cluster. Oracle strongly recommends that you use both the components. Also, monitor Oracle Clusterware-managed resources using the Clusterware resource activity log.

- [Monitoring Oracle Clusterware with Oracle Enterprise Manager](#)
Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.
- [Monitoring Oracle Clusterware with Cluster Health Monitor](#)
You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.
- [Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures](#)
The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

Related Topics

- [Managing the Cluster Resource Activity Log](#)
Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.

5.1 Monitoring Oracle Clusterware with Oracle Enterprise Manager

Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.

When you log in to Oracle Enterprise Manager using a client browser, the **Cluster Database Home** page appears where you can monitor the status of both Oracle Database and Oracle Clusterware environments. Oracle Clusterware monitoring includes the following details:

- Current and historical Cluster Health Monitor data in Oracle Enterprise Manager on the cluster target
- Notifications if there are any VIP relocations
- Status of the Oracle Clusterware on each node of the cluster using information obtained through the Cluster Verification Utility (CVU)
- Notifications if node applications (`nodeapps`) start or stop

- Notification of issues in the Oracle Clusterware alert log for the Oracle Cluster Registry, voting file issues (if any), and node evictions

The **Cluster Database Home** page is similar to a single-instance Database Home page. However, on the Cluster Database Home page, Oracle Enterprise Manager displays the system state and availability. The system state and availability includes a summary about alert messages and job activity, and links to all the database and Oracle Automatic Storage Management (Oracle ASM) instances. For example, track problems with services on the cluster including when a service is not running on all the preferred instances or when a service response time threshold is not being met.

Use the Oracle Enterprise Manager **Interconnects** page to monitor the Oracle Clusterware environment. The Interconnects page displays the following details:

- Public and private interfaces on the cluster
- Overall throughput on the private interconnect
- Individual throughput on each of the network interfaces
- Error rates (if any)
- Load contributed by database instances on the interconnect
- Notifications if a database instance is using public interface due to misconfiguration
- Throughput contributed by individual instances on the interconnect

All the information listed earlier is also available as collections that have a historic view. The historic view is useful with cluster cache coherency, such as when diagnosing problems related to cluster wait events. Access the Interconnects page by clicking the **Interconnect** tab on the Cluster Database home page.

Also, the Oracle Enterprise Manager **Cluster Database Performance** page provides a quick glimpse of the performance statistics for a database. Statistics are rolled up across all the instances in the cluster database in charts. Using the links next to the charts, you can get more specific information and perform any of the following tasks:

- Identify the causes of performance issues
- Decide whether resources must be added or redistributed
- Tune your SQL plan and schema for better optimization
- Resolve performance issues

The charts on the Cluster Database Performance page include the following:

- **Chart for Cluster Host Load Average:** The **Cluster Host Load Average** chart in the Cluster Database Performance page shows potential problems that are outside the database. The chart shows maximum, average, and minimum load values for available nodes in the cluster for the previous hour.
- **Chart for Global Cache Block Access Latency:** Each cluster database instance has its own buffer cache in its System Global Area (SGA). Using Cache Fusion, Oracle RAC environments logically combine buffer cache of each instance to enable the database instances to process data as if the data resided on a logically combined, single cache.
- **Chart for Average Active Sessions:** The **Average Active Sessions** chart in the Cluster Database Performance page shows potential problems inside the database. Categories, called wait classes, show how much of the database is using a resource, such as CPU or disk I/O. Comparing CPU time to wait time

helps to determine how much of the response time is consumed with useful work rather than waiting for resources that are potentially held by other processes.

- **Chart for Database Throughput:** The **Database Throughput** charts summarize any resource contention that appears in the Average Active Sessions chart, and also show how much work the database is performing on behalf of the users or applications. The **Per Second** view shows the number of transactions compared to the number of logons, and the amount of physical reads compared to the redo size for each second. The **Per Transaction** view shows the amount of physical reads compared to the redo size for each transaction. Logons is the number of users that are logged on to the database.

In addition, the **Top Activity** drop-down menu on the **Cluster Database Performance** page enables you to see the activity by wait events, services, and instances. In addition, you can see the details about SQL/sessions by going to a prior point in time by moving the slider on the chart.

Related Topics

- Oracle Database 2 Day + Real Application Clusters Guide

5.2 Monitoring Oracle Clusterware with Cluster Health Monitor

You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.

OCLUMON is included with Cluster Health Monitor. You can use it to query the Cluster Health Monitor repository to display node-specific metrics for a specified time period. You can also use OCLUMON to perform miscellaneous administrative tasks, such as the following:

- Changing the debug levels with the `oclumon debug` command
- Querying the version of Cluster Health Monitor with the `oclumon version` command
- Viewing the collected information in the form of a node view using the `oclumon dumpnodeview` command
- Changing the metrics database size using the `oclumon manage` command

Related Topics

- [OCLUMON Command Reference](#)
Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.

5.3 Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

If an Oracle Clusterware-managed resource fails, then Oracle Clusterware logs messages about the failure in the **cluster resource activity log** located in the Grid Infrastructure Management Repository. Failures can occur as a result of a problem

with a resource, a hosting node, or the network. The cluster resource activity log provides a unified view of the cause of resource failure.

Writes to the cluster resource activity log are tagged with an activity ID and any related data gets the same parent activity ID, and is nested under the parent data. For example, if Oracle Clusterware is running and you run the `crsctl stop clusterware -all` command, then all activities get activity IDs, and related activities are tagged with the same parent activity ID. On each node, the command creates sub-IDs under the parent IDs, and tags each of the respective activities with their corresponding activity ID. Further, each resource on the individual nodes creates sub-IDs based on the parent ID, creating a hierarchy of activity IDs. The hierarchy of activity IDs enables you to analyze the data to find specific activities.

For example, you may have many resources with complicated dependencies among each other, and with a database service. On Friday, you see that all of the resources are running on one node but when you return on Monday, every resource is on a different node, and you want to know why. Using the `crsctl query calog` command, you can query the cluster resource activity log for all activities involving those resources and the database service. The output provides a complete flow and you can query each sub-ID within the parent service failover ID, and see, specifically, what happened and why.

You can query any number of fields in the cluster resource activity log using filters. For example, you can query all the activities written by specific operating system users such as `root`. The output produced by the `crsctl query calog` command can be displayed in either a tabular format or in XML format.

The cluster resource activity log is an adjunct to current Oracle Clusterware logging and alert log messages.

**Note:**

Oracle Clusterware does not write messages that contain security-related information, such as log-in credentials, to the cluster activity log.

Use the following commands to manage and view the contents of the cluster resource activity log:

Part III

Monitoring and Managing Database Workload Performance

Oracle Database Quality of Service (QoS) Management is an automated, policy-based product that monitors the workload requests for an entire system.

- [Introduction to Oracle Database QoS Management](#)
This chapter provides an overview of Oracle Database Quality of Service Management (Oracle Database QoS Management).
- [Supported Workloads and Strategies](#)
This chapter discusses the different ways that you can use Oracle Database Quality of Service Management (Oracle Database QoS Management) to manage the workload on your system and the best practices for configuring your system to effectively use Oracle Database QoS Management.
- [Installing and Enabling Oracle Database QoS Management](#)
This chapter describes the tasks you must complete to install and configure Oracle Database QoS Management on your system. Some of the tasks in this section must be performed by the cluster administrator.
- [Administering the Oracle Database QoS Management System](#)
This chapter describes the basic administrative tasks you perform when using Oracle Database QoS Management to manage performance of your Oracle RAC cluster.
- [Troubleshooting Oracle Database QoS Management](#)
This chapter describes some problems you might encounter when using Oracle Database QoS Management and how you can resolve them. This chapter also describes how to locate the trace or log files for Oracle Database QoS Management.

Related Topics

- [Introduction to Oracle Database Quality of Service \(QoS\) Management](#)
Oracle Database Quality of Service (QoS) Management manages the resources that are shared across applications.

6

Introduction to Oracle Database QoS Management

This chapter provides an overview of Oracle Database Quality of Service Management (Oracle Database QoS Management).

- [What Is Oracle Database QoS Management?](#)
- [Benefits of Using Oracle Database QoS Management](#)
- [Overview of Oracle Database QoS Management](#)
This section provides a basic description of how Oracle Database QoS Management works, and of how it evaluates the performance of workloads on your system.
- [What Does Oracle Database QoS Management Manage?](#)
- [Overview of Metrics](#)
Oracle Database QoS Management uses a standardized set of metrics, which are collected by all the servers in the system.

6.1 What Is Oracle Database QoS Management?

Many companies are consolidating and standardizing their data center computer systems. Instead of using individual servers for each application, they run multiple applications on clustered databases. Also, the migration of applications to the Internet has introduced the problem of managing an [open workload](#). An open workload is subject to demand surges, which can overload a system, resulting in a new type of application failure that cannot be fully anticipated or planned for. To keep applications available and performing within their target service levels in this type of environment, you must:

- Pool resources.
- Have management tools that detect performance bottlenecks in real time.
- Reallocate resources to meet the change in demand.

Oracle Database QoS Management is an automated, policy-based product that monitors the workload requests for an entire system. Oracle Database QoS Management manages the resources that are shared across applications, and adjusts the system configuration to keep the applications running at the performance levels needed by your business. Oracle Database QoS Management responds gracefully to changes in system configuration and demand, thus avoiding additional oscillations in the performance levels of your applications.

Oracle Database QoS Management monitors the performance of each [work request](#) on a target system. Oracle Database QoS Management starts to track a work request from the time a work request requests a connection to the database using a database service. The amount of time required to complete a work request, or the response time (also known as the [end-to-end response time](#), or round-trip time), is the time from when the request for data was initiated and when the data request is completed. By

accurately measuring the two components of response time, which are the time spent using resources and the time spent waiting to use resources, Oracle Database QoS Management can quickly detect bottlenecks in the system. Oracle Database QoS Management then makes suggestions to reallocate resources to relieve a [bottleneck](#), thus preserving or restoring service levels.

Oracle Database QoS Management manages the resources on your system so that:

- When sufficient resources are available to meet the demand, business-level performance requirements for your applications are met, even if the workload changes.
- When sufficient resources are *not* available to meet the demand, Oracle Database QoS Management attempts to satisfy performance requirements of more business-critical workloads at the expense of less business-critical workloads.

6.2 Benefits of Using Oracle Database QoS Management

In a typical company, when the response times of your applications are not within acceptable levels, problem resolution can be very slow. Often, the first questions that administrators ask are: "Did we configure the system correctly? Is there a parameter change that fixes the problem? Do we need more hardware?" Unfortunately, these questions are very difficult to answer precisely. The result is often hours of unproductive and frustrating experimentation.

Oracle Database QoS Management provides the following benefits:

- Reduces the time and expertise requirements for system administrators who manage Oracle Real Application Clusters (Oracle RAC) resources.
- Helps reduce the number of performance outages.
- Reduces the time needed to resolve problems that limit or decrease the performance of your applications.
- Provides stability to the system as the workloads change.
- Makes the addition or removal of servers transparent to applications.
- Reduces the impact on the system caused by server failures.
- Helps ensure that service-level agreements (SLAs) are met.
- Enables more effective sharing of hardware resources.

Oracle Database QoS Management helps manage the resources that are shared by applications in a cluster. Oracle Database QoS Management can help identify and resolve performance bottlenecks. Oracle Database QoS Management does not diagnose or tune application or database performance issues. When tuning the performance of your applications, the goal is to achieve optimal performance. Oracle Database QoS Management does not seek to make your applications run faster, but instead works to remove obstacles that prevent your applications from running at their optimal performance levels.

6.3 Overview of Oracle Database QoS Management

This section provides a basic description of how Oracle Database QoS Management works, and of how it evaluates the performance of workloads on your system.

- [How Does Oracle Database QoS Management Work?](#)

- [Overview of Policy Sets](#)
- [Overview of Server Pools](#)
- [How Server Pools Are Used by Oracle Database QoS Management](#)
You should create one or more server pools depending on the workloads that need to be managed.
- [Overview of Performance Classes](#)
- [Overview of Performance Policies and Performance Objectives](#)
To manage the various Performance Objectives, you define one or more Performance Policies.
- [How Oracle Database QoS Management Collects and Analyzes Performance Data](#)
The Oracle Database QoS Management Server retrieves metrics data from Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node databases in the cluster.
- [Overview of Recommendations](#)
Oracle Database QoS Management enables you to manage excess capacity to meet specific performance goals through its recommendations.

6.3.1 How Does Oracle Database QoS Management Work?

With Oracle Database, you can use services to manage the workload on your system by starting services on groups of servers that are dedicated to particular workloads. At the database tier, for example, you could dedicate one group of servers to online transaction processing (OLTP), dedicate another group of servers to application testing, and dedicate a third group of servers for internal applications. The system administrator can allocate resources to specific workloads by manually changing the number of servers on which a database service is allowed to run.

Using groups of servers in this way isolates the workloads from each other to prevent demand surges, failures, and other problems in one workload from affecting the other workloads. However, in this type of deployment, you must separately provision the servers to each group to satisfy the peak demand of each workload because resources are not shared.

Oracle Database QoS Management performs the following actions:

1. Uses a policy created by the QoS administrator to do the following:
 - Assign each work request to a [Performance Class](#) by using the attributes of the incoming work requests (such as the database service to which the application connects).
 - Determine the target response times ([Performance Objectives](#)) for each Performance Class.
 - Determine which Performance Classes are the most critical to your business
2. Monitors the resource usage and resource wait times for all the Performance Classes.
3. Analyzes the average response time for a Performance Class against the Performance Objective in effect for that Performance Class
4. Produces recommendations for reallocating resources to improve the performance of a Performance Class that is exceeding its target response time, and provides an

analysis of the predicted impact to performance levels for each Performance Class if that recommendation is implemented.

5. Implements the actions listed in the recommendation when directed to by the Oracle Database QoS Management administrator, then evaluates the system to verify that each Performance Class is meeting its Performance Objective after the resources have been reallocated.
- [Oracle Database QoS Management and Server Pools](#)
You can use server pools to create groups of servers within a cluster to provide workload isolation.
 - [Oracle Database QoS Management and Instance Caging](#)
 - [Oracle Database QoS Management and Services](#)
Oracle Database QoS Management uses database services to monitor and manage client and application workloads.

6.3.1.1 Oracle Database QoS Management and Server Pools

You can use server pools to create groups of servers within a cluster to provide workload isolation.

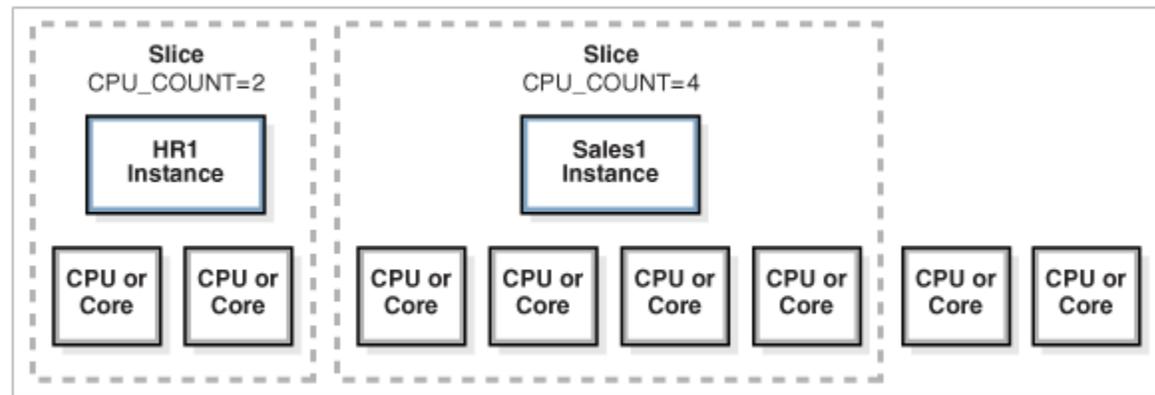
A server can only belong to one server pool at any time. You can create an Oracle Database in a single server pool, or across multiple server pools. Oracle Database QoS Management can make recommendations to move a server from one server pool to another based on the measured and projected demand. Oracle Database QoS Management can also relocate servers to satisfy the Performance Objectives currently in effect.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

6.3.1.2 Oracle Database QoS Management and Instance Caging

When multiple database instances share a single server, they must share its CPU, memory, and I/O bandwidth. Instance Caging limits the amount of CPU an Oracle database instance consumes by using the Oracle Database Resource Manager and the `CPU_COUNT` database initialization parameter. When using Oracle Database QoS Management, the sum of the values for `CPU_COUNT` for all instances of the server must be less than or equal to the total number of physical CPUs. Also, each CPU partition, or **slice**, must be uniform in thickness (number of CPUs) for each instance of a database within a server pool. These requirements help to ensure predictable and isolated performance of each database.

Figure 6-1 Instance Caging and CPU Slices

When you implement instance caging, Oracle Database QoS Management can provide recommendations to reallocate CPU resources from one slice to another slice within the same server pool. If you choose to implement the recommendation to modify the instance caging settings, then Oracle Database QoS Management modifies the `CPU_COUNT` parameter uniformly for all the database instances running on servers in the server pool.

Modifying the `CPU_COUNT` parameter and configuring Oracle Database QoS Management so that a Resource Plan is activated enables the Instance Caging feature. When you use Instance Caging to constrain CPU usage for an instance, that instance could become CPU-bound. This is when the Resource Manager begins to do its work, allocating CPU shares among the various database sessions according to the active resource plan.

6.3.1.3 Oracle Database QoS Management and Services

Oracle Database QoS Management uses database services to monitor and manage client and application workloads.

In an Oracle RAC cluster, Oracle Database QoS Management monitors the server pools and nodes on which the database services are offered. A service can run in only one server pool. If the database spans multiple server pools, then you must create multiple services to access the instances in all server pools.

In the case of Administrator-Managed databases, all instances run in the Generic server pool and services run on specific instances, but still within a single pool.

Workload is monitored for clients and applications that connect to the database using database services that are managed by Oracle Clusterware. The connections must use Java Database Connectivity (JDBC) (thick or thin), or Oracle Call Interface (OCI). Connections should use services with its run-time goal for the load balancing advisory set to `SERVICE_TIME`, and the connection load balancing goal set to `LONG`. For example:

```
srvctl modify service -db db_name -service service_name -rlbgoal SERVICE_TIME
-clbgoal LONG -cardinality UNIFORM
```

You must define the cardinality of database services as follows:

- If the server pool that the service runs in has a maximum size greater than 1 (or UNLIMITED), then set the cardinality of the service to `UNIFORM`.
- If the server pool that the service runs in has a maximum size of 1, then set the cardinality of the service to `SINGLETON`.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

6.3.2 Overview of Policy Sets

The central concept in Oracle Database QoS Management is the [Policy Set](#). A Policy Set enables you to specify your resources, Performance Classes (workloads), and one or more Performance Policies that specify the Performance Objective for each Performance Class. A Policy Set can also specify constraints for resource availability. Oracle Database QoS Management Performance Policies manage the availability of resources system wide for each Performance Class so that the system is able to satisfy the Performance Objectives you set in the Performance Policy.

When you use Oracle Enterprise Manager to create a new Default Policy for your system, Oracle Database QoS Management provides default classification rules and associated Performance Class names. For example, when you create the initial Policy Set, Oracle Database QoS Management discovers all database services in a cluster, and creates a Performance Class for each service. The Performance Class is named by appending `_pc` to the service name. For example, if the name of a service is `sales`, then the name assigned to the Performance Class for that service is `sales_pc`.

Only one Performance Policy in the Policy Set can be active at any time. You can activate Performance policies to respond to particular requirements by using calendar schedules, maintenance windows, events, and so on.

When you create a Policy Set, you specify which server pools in the cluster should be managed by Oracle Database QoS Management. You also define Performance Classes (used to categorize workloads with similar performance requirements). You then create a Performance Policy to specify which Performance Classes have the highest priority and the Performance Objectives of each Performance Class. To satisfy the Performance Objectives, Oracle Database QoS Management makes recommendations for reallocating resources when needed, and predicts what effect the recommended actions will have on the ability of each Performance Class to meet its Performance Objective.

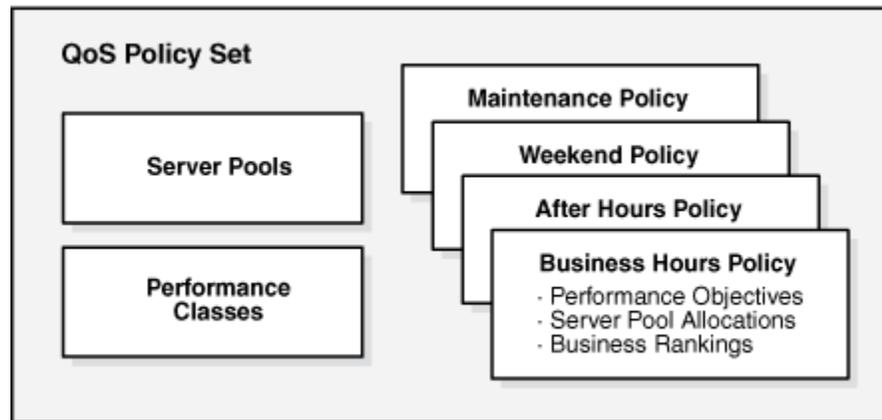
For example, you could create a policy to manage your application workloads during business hours. The applications used by customers to buy products or services are of the highest priority to your business during this time. You also give high priority to order fulfillment and billing applications. Human resource and enterprise resource planning (ERP) applications are of a lower priority during this time. If your online sales applications experience a surge in demand, then Oracle Database QoS Management might recommend that more resources be allocated to the sales applications and taken away from applications of lesser importance. The recommendation also includes a prediction of the change in performance (positive or negative) for each Performance Class.

A Policy Set, as shown in [Figure 6-2](#), consists of the following:

- The server pools that are being managed by Oracle Database QoS Management

- Performance Classes, which are groups of work requests with similar performance objectives
- Performance policies, which describe how resources should be allocated to the Performance Classes by using:
 - [Performance Objectives](#)
 - [Performance Class ranks](#)
 - [server pool directive overrides](#)

Figure 6-2 Elements of an Oracle Database QoS Management Policy Set



Related Topics

- [Overview of Performance Policies and Performance Objectives](#)
 To manage the various Performance Objectives, you define one or more Performance Policies.
- [Overview of Recommendations](#)
 Oracle Database QoS Management enables you to manage excess capacity to meet specific performance goals through its recommendations.
- [Applying Classifiers to Work Requests](#)
- [Create an Initial Policy Set](#)
- [Administering the Policy Set](#)
 Whether you are configuring the Oracle Database QoS Management system for the first time, or want to create a new Policy Set, you use the Create Policy Set wizard to create your Policy Set. You use the Edit Policy Set wizard to modify your existing Policy Set.

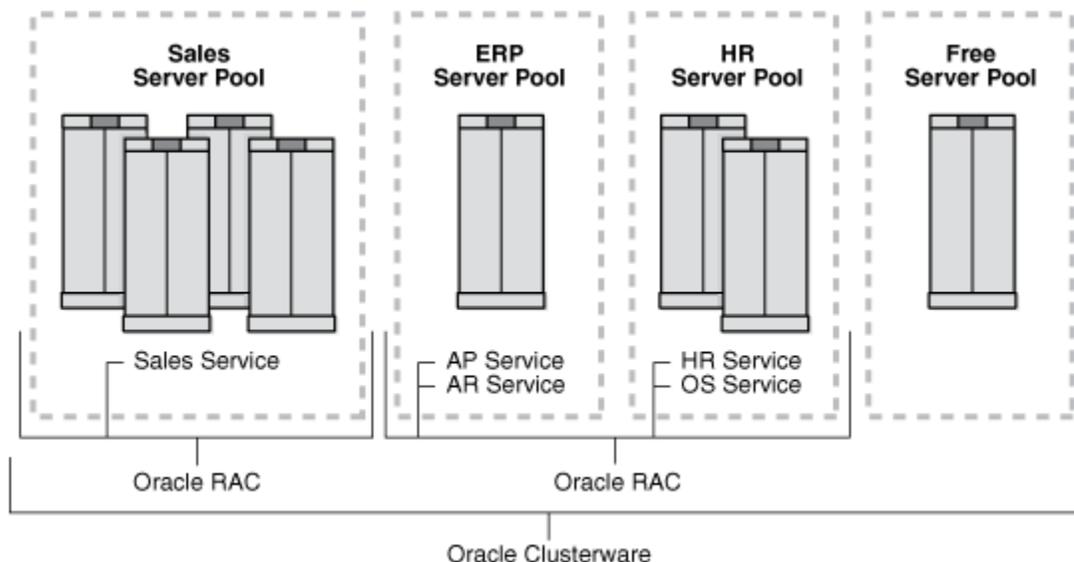
6.3.3 Overview of Server Pools

When deciding how many clusters to create for your business, you need to compare the possible cost savings through consolidation of servers with the risk that the consolidated workloads will interfere with each other in some significant way. With the introduction of server pools to logically divide a cluster, you can achieve the benefit of physical consolidation and resource agility while maintaining workload isolation.

As the administrator, you can define the workloads that can run in various server pools, as shown in [Figure 6-3](#). Applications that connect to your Oracle RAC database

use a service that runs only on the servers currently allocated to that server pool. For example, in [Figure 6-3](#), connections and applications that use the OS service access only the servers in the HR server pool, so that work done by those connections QoS does not interfere with the applications using the Sales service. Oracle Database QoS Management can assist you with managing the resource allocations within each of those groups to meet your service levels, and can redistribute resources automatically to meet changes in your business requirements.

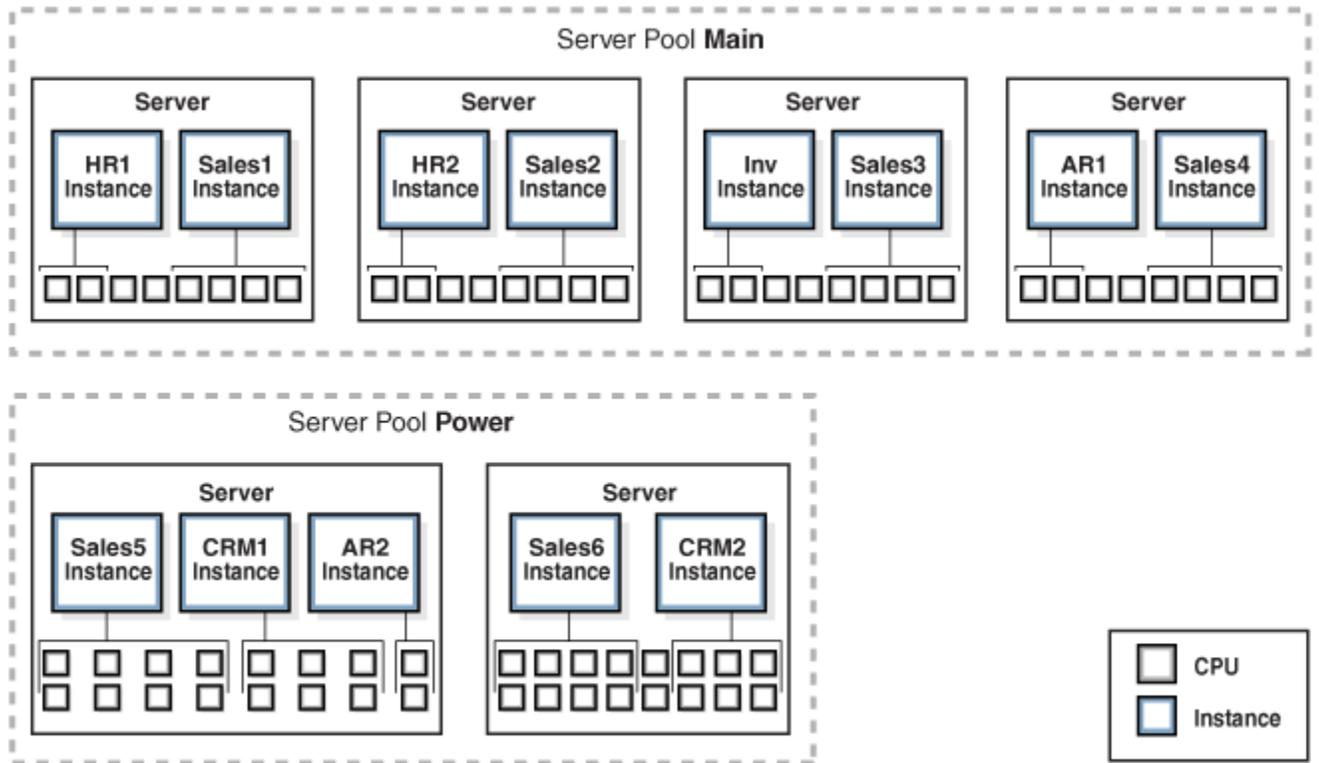
Figure 6-3 Diagram of Server Pools, Oracle Databases, and Database Services



With server pools, you can now create groups of servers that can be managed as a single entity. Databases can be created to run in these server pools. If each server runs only a single instance of a database, then if the database needs more resources, an additional server can be allocated to the server pool. If multiple database instances run on a single server, then they must compete for the shared resources of that server, such as memory and CPU. If one of the database instances is experiencing a much higher workload than the other instances, then that database instance can significantly degrade the performance of the other instances running on the same server.

You can use Instance Caging to limit the amount of CPU an Oracle Database instance consumes. By setting the `CPU_COUNT` parameter to limit the maximum number of CPUs an instance can use, you partition the CPUs among the database instances on a server, thus preventing them from using excessive amounts of CPU resources. The `CPU_COUNT` setting must be the same for each instance of a database within a server pool. Oracle Database QoS Management can monitor the CPU usage among all the database instances in the server pool, and recommend changes to the current settings if needed.

Figure 6-4 Server Pools and CPU Slices



Instead of using a single server pool, Oracle recommends that you manage services by creating multiple server pools, and relocate services between them. Using this configuration provides the following benefits:

- Different types of workloads require different configurations, and have different tuning goals. For example, a customer using your OLTP applications to purchase goods or services expects the shipping and payment information screens to respond quickly. If your applications take too long to process the order, then the customer can lose interest, and your company might lose a sale. By contrast, an employee using an internal HR application is motivated to continue using HR screens, even if they do not respond quickly. If your HR applications take longer than expected to process an online task, the employee is unlikely to quit.
- Applications can have various resource requirements throughout the day, week, or month to meet your business objectives. You can use server pools to divide the resources among the application workloads. To meet the Performance Objectives of a given time period, you could use server pool directive overrides in a Performance Policy to change the default attributes (such as Max or Min) for a server pool.

For example, if your company has an online tax filing application, then the application must prepare and file the tax statements for your customers before the government-specified deadline. In the timeframe immediately preceding a filing deadline, applications related to tax statement preparation and filing require more resources than they do at other times of the year. To ensure you meet this service requirement, you can create a Performance Policy named QuarterlyFilings to override the standard server pool directive, and specify that when QuarterlyFilings

is active, the server pool used by the tax preparation applications should have a minimum of four servers instead of two to handle the additional workload. When the Quarterly Filings Performance Policy is not in effect, your default Performance Policy is in effect, and the minimum number of servers in that server pool is two.

- Because Oracle Database QoS Management regulates the number of servers that support a workload, application users experience a consistent level of performance, even in the presence of changing demand levels. This prevents performance expectations of your customers from being reset when workload levels change from low to high demand.

For example, assume your company sells a new consumer product that is in high demand, and your company advertises that they have large quantities of that product for sale at a reduced price. As a result, many new customers create orders for this product, and your OLTP applications must process a rapidly increasing number of transactions (a demand surge occurs). New customers do not know what to expect with regards to the OLTP application performance. However, existing customers can react negatively if their online shopping experience is impacted by the flood of new customers. Also, if your OLTP application cannot process all the incoming orders, then some of the new customers might quit the application and place their order with a different company, or visit a retail store instead.

Oracle Database QoS Management helps you to manage the reallocation of available resources to meet the demand surge without sacrificing the quality of service of your other applications.

- Some workloads do not scale well but still benefit from the high availability of a cluster environment. Deploying these workloads in a fixed-size server pool offers both performance manageability and high availability.

For example, if you run an ERP application in a server pool with a fixed size of one server, then the maximum size of the server pool and the minimum size of the server pool are both set to one. If the server in that server pool fails, then Oracle Clusterware automatically allocates a new server to the server pool to maintain the minimum size of one server. Any instances and services located on the failed server are restarted on the new server, so the applications using these instances and services remains available.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about server pools
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about policy-managed Oracle RAC databases
- *Oracle Database Concepts* for a description of an OLTP system

6.3.4 How Server Pools Are Used by Oracle Database QoS Management

You should create one or more server pools depending on the workloads that need to be managed.

When you first install Oracle Grid Infrastructure for a cluster, a default server pool (the **Free pool**) is created. All servers are initially placed in this server pool. When you create a new server pool, the servers that you assign to that server pool are automatically moved out of the Free pool and placed in the newly created server pool. At this point, you can install a database to run in that server pool, and create database services that are managed by Oracle Clusterware for applications to connect to that database.

For an Oracle RAC database to take advantage of the flexibility of server pools, the database must be created using the policy-managed deployment option, which places the database in one or more server pools.

In Oracle Database 12c release 1 (12.1.0.2), Oracle Database Quality of Service (QoS) Management supported administrator-managed Oracle RAC and Oracle RAC One Node databases in Measure-Only and Monitor modes. Starting with Oracle Database 12c release 2 (12.2.0.1), you can use Oracle Database QoS Management in Management mode with Oracle RAC and Oracle RAC One Node databases that are policy-managed or administrator-managed.

Caution:

If you use candidate server lists (`server_names` attribute) or categories when creating server pools, then the ability for Oracle Database Quality of Service (QoS) Management to expand the server pool will be limited by those restrictions as non-eligible servers cannot be used.

Related Topics

- [Oracle Real Application Clusters Installation Guide for Linux and UNIX](#)
- [Oracle Real Application Clusters Administration and Deployment Guide](#)

6.3.5 Overview of Performance Classes

A Policy Set contains Performance Objectives for various Performance Classes, or workloads, that run on your cluster. Oracle Database QoS Management uses a set of classification rules defined in the Policy Set to categorize work requests into a Performance Class. The fundamental classifier used to assign work requests to Performance Classes is the name of the service that is used to connect to the database.

- [Performance Class Tags](#)
- [Applying Classifiers to Work Requests](#)
- [Using Additional Filters for Classifying Work Requests](#)
- [Deciding to Create New Performance Classes](#)
Over time, your workload and performance goals can change.

6.3.5.1 Performance Class Tags

The classification of work requests applies a user-defined name (**tag**) that identifies the Performance Class to which the work request belongs. All work requests that are grouped into a particular Performance Class have the same **performance objectives**.

In effect, the tag connects the work request to the Performance Objective for the associated Performance Class. Oracle Database QoS Management assigns tags to each work request so that every component of the system can take measurements, and provide data to Oracle Database QoS Management for evaluation against the applicable Performance Objectives.

6.3.5.2 Applying Classifiers to Work Requests

Classification occurs wherever new work enters the system. When a work request arrives at a server, the work request is checked for a tag. If the work request has a tag, then the server concludes that this work request has already been classified, and the tag is not changed. If the work request does not include a tag, then the [classifiers](#) are checked, and a tag for the matching Performance Class is attached to the work request.

To illustrate how work requests are classified, consider an application that connects to an Oracle RAC database. The application uses the database service `sales`. The Oracle Database QoS Management administrator specified during the initial configuration of Oracle Database QoS Management that the `sales_pc` Performance Class should contain work requests that use the `sales` service. When a connection request is received by the database, Oracle Database QoS Management checks for a tag. If a tag is not found, then Oracle Database QoS Management compares the information in the connection request with the classifiers specified for each Performance Class, in the order specified in the Performance Policy. If the connection request being classified is using the `sales` service, then when the classifiers in the `sales_pc` Performance Class are compared to the connection request information, a match is found, and the database work request is assigned a tag for the `sales_pc` Performance Class.

6.3.5.3 Using Additional Filters for Classifying Work Requests

A single application can support work requests of many types, with a range of performance characteristics. By extending and refining the default classification rules, the Oracle Database QoS Management administrator can write multiple Performance Objectives for a single application. For example, the administrator may decide that a web-based application should have separate Performance Objectives for work requests related to logging in, browsing, searching, and purchasing.

Oracle Database QoS Management supports user-defined combinations of connection parameters to map Performance Classes to the actual workloads running in the database. These connection parameters belong to two general classes, and can be combined to create fine-grained Boolean expressions:

- **Configuration Parameters**—The supported configuration parameters are `SERVICE_NAME` and `USERNAME`. Each classifier in a Performance Class must specify the name of a database service. Additional granularity can be achieved by identifying the name of the user that is making the database connection from either a client or the middle tier. The advantage of using these classifiers is that they do not require application code changes to associate different workloads with separate Performance Classes.
- **Application Parameters**—The supported application parameters are `MODULE`, `ACTION`, and `PROGRAM`. These are optional parameters. The values for `MODULE` and `ACTION` must be set within the application. Depending on the type of application, you can set these parameters as follows:

- OCI—Use `OCI_ATTR_MODULE` and `OCI_ATTR_ACTION`.
- Oracle Data Provider for .NET (ODP.NET)—Specify the `ModuleName` and `ActionName` properties on the `OracleConnection` object.
- JDBC—Set `MODULE` and `ACTION` in `SYS_CONTEXT`.

The `PROGRAM` parameter is set or derived differently for each database driver and platform. Consult the appropriate Oracle Database developer's guide for further details and examples.

To manage the workload for an application, the application code makes database connections using a particular service. To provide more precise control over the workload generated by various parts of the application, you can create additional Performance Classes, and use classifiers that include `PROGRAM`, `MODULE`, or `ACTION` in addition to the service or user name. For example, you could specify that all connections to your cluster that use the `sales` service belong to the `sales_pc` Performance Class, but connections that use the `sales` service and have a user name of `APPADMIN` belong to `sales_admin` Performance Class.

Related Topics

- *Oracle Call Interface Programmer's Guide*
- *Oracle Data Provider for .NET Developer's Guide for Microsoft Windows*
- *Oracle Database JDBC Developer's Guide*

6.3.5.4 Deciding to Create New Performance Classes

Over time, your workload and performance goals can change.

The Performance Classes in use at a particular data center are expected to change over time. For example, you might need to modify the Performance Objectives for one part of your application. In this case you would create a new Performance Class with additional classifiers to identify the target work requests, and update your Performance Policy to add a new Performance Objective for this Performance Class. In other words, you replace a single Performance Objective with one or more finer-grained Performance Objectives, and divide the work requests for one Performance Class into multiple Performance Classes.

Application developers can suggest which Performance Classes to use. Specifically, an application developer can suggest ways to identify different application workloads, and you can use these suggestions to create classifiers for Performance Classes so that each type of work request is managed separately.

You can create additional Performance Classes to specify acceptable response times for different application workloads. For example, a Performance Objective might indicate that a work request performing the `checkout` action for the `sales_pc_checkout` Performance Class should not take more than one millisecond to complete, and a work request performing the `browse` action for the `sales_pc_browse` Performance Class can take 100 milliseconds second to complete.

Related Topics

- [Managing Performance Classes](#)

6.3.6 Overview of Performance Policies and Performance Objectives

To manage the various Performance Objectives, you define one or more Performance Policies.

A [Performance Policy](#) is a collection of Performance Objectives, and a measure of how critical they are to your business. For example, you could define a Performance Policy for normal business hours, another for weekday nonbusiness hours, one for weekend operations, and another to be used during processing for the quarter-end financial closing. At any given time, a single Performance Policy is in effect as specified by the Oracle Database QoS Management administrator. Within each Performance Policy, the criticalness, or ranking, of the Performance Objectives can be different, enabling you to give more priority to certain workloads during specific time periods.

A Performance Policy has a collection of Performance Objectives in effect at the same time; there is one or more Performance Objectives for each application or workload that runs on the cluster. Some workloads and their Performance Objectives are more critical to the business than others. Some Performance Objectives can be more critical at certain times, and less critical at other times.

- [Overview of Performance Objectives](#)
You create Performance Objectives for each Performance Class to specify the target performance level for all work requests that are assigned to each Performance Class.
- [Overview of Server Pool Directive Overrides](#)
A server pool directive override sets the availability properties of Min, Max, and Importance for a server pool when the Performance Policy is in effect.
- [Overview of Performance Class Ranks](#)
Specifying a rank for Performance Classes helps to prioritize work.
- [Oracle Database QoS Management Policy Workload Criticality Determines Database Startup Order](#)

6.3.6.1 Overview of Performance Objectives

You create Performance Objectives for each Performance Class to specify the target performance level for all work requests that are assigned to each Performance Class.

[Performance Objectives](#) specify both the business requirement (the target performance level) and the work to which that Performance Objective applies (the Performance Class). For example, a Performance Objective could specify that work requests in the `hr_pc` Performance Class should have an average response time of less than 0.2 seconds.

Performance Objectives are specified with Performance Policies. Each Performance Policy includes a Performance Objective for each and every Performance Class, unless the Performance Class is marked Measure-Only. In this release, Oracle Database QoS Management supports only one type of Performance Objective, [average response time](#).

The response time for a workload is based upon database client requests. Response time measures the time from when the cluster receives the request over the network to the time the request leaves the cluster. Response time does not include the time required to send the information over the network to or from the client. The response

time for all database client requests in a Performance Class is averaged and presented as average response time, measured in seconds for a database request.

Related Topics

- [Creating a Performance Policy and Specifying Performance Objectives](#)
Use Oracle Enterprise Manager Cloud Control to create a Performance Policy.

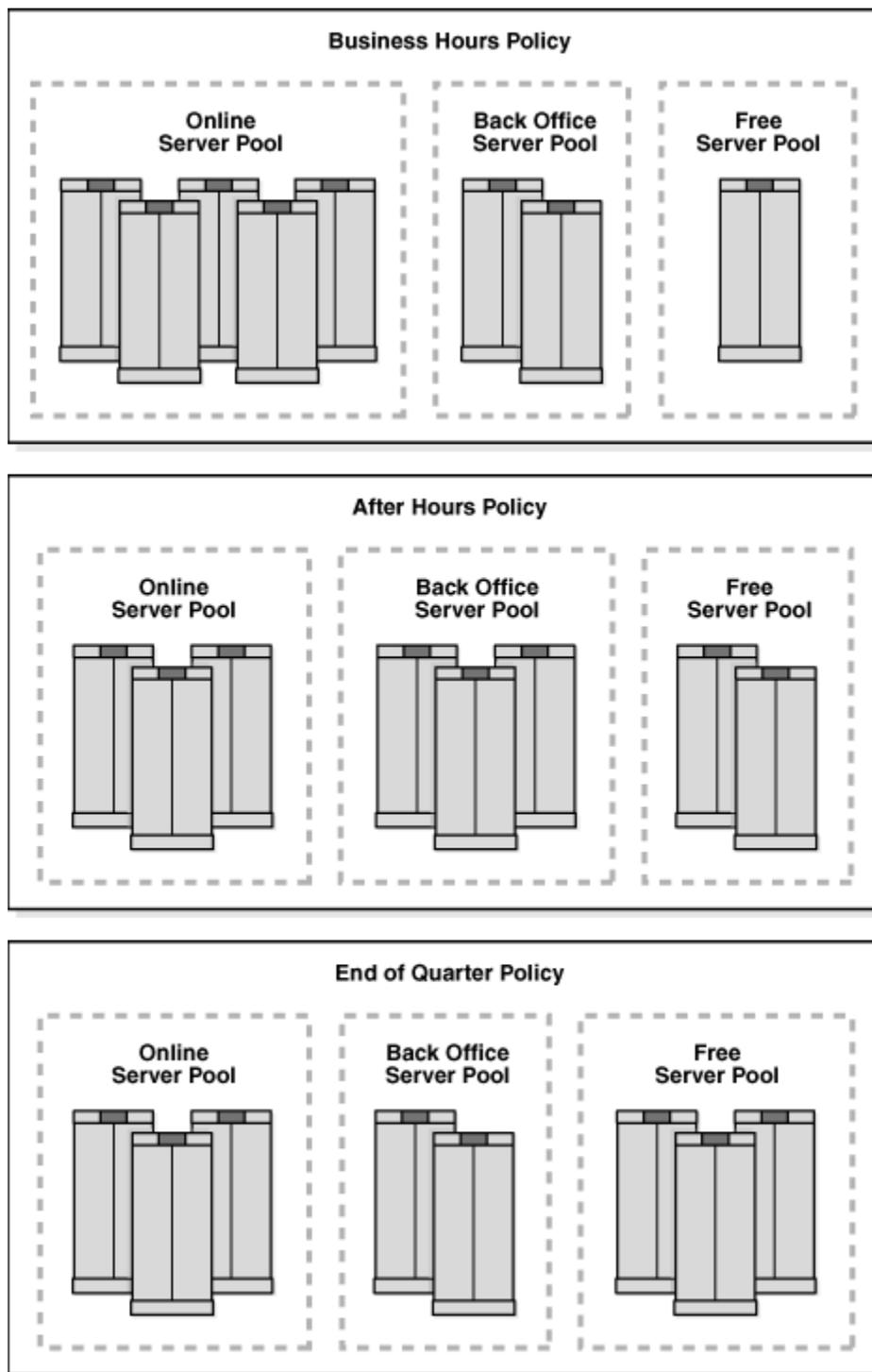
6.3.6.2 Overview of Server Pool Directive Overrides

A server pool directive override sets the availability properties of Min, Max, and Importance for a server pool when the Performance Policy is in effect.

A Performance Policy can include a set of [server pool directive overrides](#). Server pool directive overrides serve as constraints on the allocation changes that Oracle Database QoS Management recommends, because the server pool directive overrides are honored during the activation period of the Performance Policy. For example, Oracle Database QoS Management never recommends moving a server out of a server pool if doing so results in the server pool having less than its specified minimum number of servers.

You could create Performance Policies for your system to manage workload based on the time of year or time of day, as shown in [Figure 6-5](#). Under normal conditions, these Performance Policies keep your database workload running at a steady rate. If the workload requests for a database increase suddenly, then a particular server pool might require additional resources beyond what is specified by the Performance Policy.

Figure 6-5 Baseline Resource Management by Performance Policy



For example, assume your business takes orders over the telephone, and creates orders using a sales application. Your telephone sales department is only open during regular business hours, but customers can also place orders themselves over the Internet. During the day, more orders are placed so the sales applications need more resources to handle the workload. This configuration is managed by creating the

Business Hours Performance Policy, and specifying that the Back Office server pool can have a maximum of two servers, enabling Oracle Database QoS Management to move servers to the Online server pool, as needed. After the telephone sales department closes, the workload for the sales applications decreases. To manage this configuration you create the After Hours Performance Policy and specify that the Back Office server pool can have a maximum of four servers, enabling your internal applications to acquire the additional resources that they need to complete their workloads before the next business day.

In this scenario, the Business Hours and After Hours Performance Policies can contain server pool directive overrides. When a Performance Policy contains a server pool directive override, the current settings of Max, Min, and Importance for the specified server pool are overridden while that Performance Policy is in effect. This enables additional servers to be placed in the Sales server pool to give the online sales applications the resources they need and to limit the resources used by the Back Office server pool, so that its workload does not interfere with the Sales workload.

Related Topics

- [Setting Server Pool Directive Overrides](#)
A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.

6.3.6.3 Overview of Performance Class Ranks

Specifying a rank for Performance Classes helps to prioritize work.

Within a Performance Policy, you can assign a level of business criticalness (a *rank*) to each Performance Class to give priority to meeting the Performance Objectives for a more critical Performance Class over a less critical one. When there are not enough resources available to meet all the Performance Objectives for all Performance Classes at the same time, the Performance Objectives for the more critical Performance Classes must be met at the expense of the less critical Performance Objectives. The Performance Policy specifies the *business criticalness* of each Performance Class, which can be Highest, High, Medium, Low, or Lowest.

 **Note:**

Priority access to resources, based on rank, does not apply to single-instance Oracle RAC databases or Oracle RAC One Node.

For example, using the Performance Policies illustrated in [Figure 6-5](#), when the Business Hours Performance Policy is in effect, the sales applications, which access the Online server pool, have the highest rank. If there are not enough resources available to meet the Performance Objectives of all the Performance Classes, then the applications that use the Online server pool will get priority access to any available resources, even if the applications using the Back Office server pool are not meeting their Performance Objectives.

You can have multiple Performance Classes at the same rank. If Oracle Database QoS Management detects more than one Performance Class not meeting its Performance Objective and the Performance Classes are assigned the same rank in the active Performance Policy, then Oracle Database QoS Management recommends a change to give the Performance Class closest to meeting its Performance Objective

more resources. After implementing the recommended action, when the Performance Class is no longer below its target performance level, Oracle Database QoS Management performs a new evaluation of the system performance.

Related Topics

- [Managing Performance Policies](#)

6.3.6.4 Oracle Database QoS Management Policy Workload Criticality Determines Database Startup Order

If a user-created Oracle Database QoS Management policy is active, then the ranked order of the performance classes determines the order in which the associated Oracle RAC databases start or request real-time LMS process slots. Using the performance class rankings ensures that mission critical databases running in a consolidated environment have their LMS processes run in real-time, thus eliminating a resource bottleneck within inter-node communication. Because the Oracle Database QoS Management policy specifies the rank of each workload, using the value of `Max(Ranks)` for each database provides a consistent expression of the expressed business criticality of each database.

6.3.7 How Oracle Database QoS Management Collects and Analyzes Performance Data

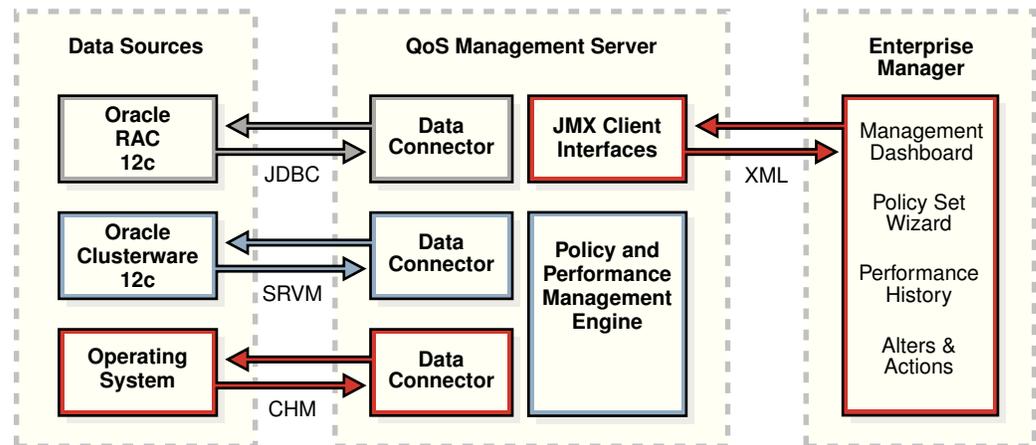
The Oracle Database QoS Management Server retrieves metrics data from Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node databases in the cluster.

The data are correlated by Performance Class every five seconds. The data include many metrics such as database request arrival rate, CPU use, CPU wait time, I/O use, I/O wait time, Global Cache use and Global Cache wait times. Information about the current topology of the cluster and the health of the servers is added to the data. The Policy and Performance Management engine of Oracle Database QoS Management (illustrated in [Figure 6-6](#)) analyzes the data to determine the overall performance profile of the system for the current Performance Objectives established by the active Performance Policy.

The performance evaluation occurs once a minute, and results in a recommendation if any Performance Class does not meet its objectives. The recommendation specifies which resource is the **bottleneck**. Specific corrective actions are included in the recommendation, if possible. The recommendation also includes a listing of the projected impact on all Performance Classes in the system if you decide to implement the recommended action.

[Figure 6-6](#) diagrams the collection of data from various data sources and shows how that information is used by Oracle Enterprise Manager. In this figure, CHM refers to Oracle Cluster Health Monitor and Server Manager (SRVM) is a component of Oracle Clusterware.

Figure 6-6 Diagram of Oracle Database QoS Management Server Architecture



Related Topics

- [Using the Oracle Database QoS Management Dashboard](#)
 The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.
- [Interpreting the Performance Overview Graphs](#)
- [Reviewing Performance Metrics](#)
 You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

6.3.8 Overview of Recommendations

Oracle Database QoS Management enables you to manage excess capacity to meet specific performance goals through its recommendations.

If your business experiences periodic demand surges or must support an [open workload](#), then to retain performance levels for your applications you can design your system to satisfy the peak workload. Creating a system capable of handling the peak workload typically means acquiring additional hardware to be available when needed and sit idle when not needed. Instead of having servers remain idle except when a demand surge occurs, you could decide to use those servers to run other application workloads. However, if the servers are busy running other applications when a demand surge hits, then your system might not be able to satisfy the peak workload and your main business applications do not perform as expected.

- [How Oracle Database QoS Management Generates Recommendations](#)
 If a performance objective is not being met, then Oracle Database QoS Management evaluates several possible solutions and presents the most effective solutions as Recommendations.
- [Types of Recommendations](#)
 If Performance Objectives are not being met for a Performance Class, then Oracle Database Quality of Service Management issues recommendations to rebalance the use of resources to alleviate bottlenecks.

- [Choosing the Best Recommendation](#)
Oracle Database Quality of Service Management can offer multiple recommendations for improving workload performance.
- [Contents of a Recommendation](#)
Each recommendation is comprised of several pieces of information.
- [Overview of Implementing Recommendations](#)
Oracle Database QoS Management does not implement the recommendations automatically.
- [Example: How Recommendations Are Generated](#)

6.3.8.1 How Oracle Database QoS Management Generates Recommendations

If a performance objective is not being met, then Oracle Database QoS Management evaluates several possible solutions and presents the most effective solutions as Recommendations.

When you use Oracle Database QoS Management, your system is continuously monitored in an iterative process to see if the Performance Objectives in the active Performance Policy are being met. Performance data are sent to Oracle Enterprise Manager for display in the Oracle Database QoS Management Dashboard (the Dashboard) and Performance History pages.

When one or more Performance Objectives are not being met, after evaluating the performance of your system, Oracle Database QoS Management seeks to improve the performance of a single Performance Objective: usually the highest ranked Performance Objective that is currently not being satisfied. If all Performance Objectives are satisfied with capacity to spare for both the current and projected workload, then Oracle Database QoS Management signals "No action required: all Performance Objectives are being met."

Related Topics

- [Using the Oracle Database QoS Management Dashboard](#)
The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.
- [Viewing Recommendations](#)
- [Reviewing Performance Metrics](#)
You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

6.3.8.2 Types of Recommendations

If Performance Objectives are not being met for a Performance Class, then Oracle Database Quality of Service Management issues recommendations to rebalance the use of resources to alleviate bottlenecks.

Oracle Database QoS Management evaluates several possible solutions and then chooses the solution that:

- Offers the best overall system improvement
- Causes the least system disruption
- Helps the highest ranked violating performance class

The types of recommendations that Oracle Database QoS Management can make are:

- [Promoting and Demoting Consumer Groups](#)
 - [Modifying the CPU Count](#)
 - [Moving Servers Between Server Pools](#)
 - [Modifying CPU Shares Assigned to Pluggable Databases](#)
- Oracle Database Quality of Service (QoS) Management manages the CPU resources used across pluggable databases (PDBs) in a multitenant database.

6.3.8.2.1 Promoting and Demoting Consumer Groups

If Performance Objectives are not being met for a Performance Class, and the Performance Class accesses the same database as other Performance Classes, then Oracle Database QoS Management can recommend consumer group mapping changes. Changing the consumer group mappings gives more access to the CPU resource to the Performance Class that is not meeting its Performance Objective. Oracle Database QoS Management issues consumer group mapping recommendations only for Performance Classes that are competing for resources in the same database and server pool.

6.3.8.2.2 Modifying the CPU Count

If you have multiple database instances running on servers in a server pool, Oracle Database QoS Management can recommend that CPU resources used by a database instance in one slice on the server be donated to a slice that needs more CPU resources. If there is a Performance Class that is not meeting its Performance Objective, and there is another slice on the system that has available headroom, or the Performance Classes that use that slice are of a lower rank, then Oracle Database QoS Management can recommend moving a CPU from the idle slice to the overloaded slice. If this recommendation is implemented, then the `CPU_COUNT` parameter is adjusted downwards for the idle instance and upwards for the overworked instance on all servers in the server pool.

6.3.8.2.3 Moving Servers Between Server Pools

Another recommended action that Oracle Database QoS Management can display is to move a server from one server pool to another to provide additional resources to meet the Performance Objectives for a Performance Class. If all the server pools in the cluster are at their specified minimum size, or if the server pool needing the resource is at its maximum size, then Oracle Database QoS Management can no longer recommend removing servers from server pools. In this situation the Dashboard displays "No recommended action at this time."

 **Note:**

This type of recommendation is not available for administrator-managed databases.

The minimum size of a server pool is the number of servers that that server pool is required to have. If you add the values for the server pool minimum attribute for each

server pool in your cluster, then the difference between this sum and the total number of servers in the cluster represents shared servers that can move between server pools (or *float*) to meet changes in demand. For example, if your cluster has 10 servers and two server pools, and each server pool has a minimum size of four, then your system has two servers that can be moved between server pools. These servers can be moved if the target server pool has not reached its maximum size. Oracle Database QoS Management always honors the Min and Max size constraints set in a policy when making Move Server recommendations.

If you set the minimum size of a server pool to zero and your system experiences a demand surge, then Oracle Database QoS Management can recommend moving all the servers out of that server pool so that the server pool is at its minimum size. This results in the Performance Classes that use that server pool being completely starved of resources, and essentially being shut down. A server pool with a minimum size of zero should only host applications that are of low business criticalness and Performance Classes that are assigned a low rank in the Performance Policy.

6.3.8.2.4 Modifying CPU Shares Assigned to Pluggable Databases

Oracle Database Quality of Service (QoS) Management manages the CPU resources used across pluggable databases (PDBs) in a multitenant database.

Each pluggable database is managed independently. If Performance Objectives are not being met for a Performance Class that utilizes a pluggable database, then Oracle Database QoS Management can recommend that the CPU shares assigned to the pluggable database be increased. The assignment of CPU shares is implemented through Database Resource Manager consumer group mappings in the resource plan. When a CDB is managed, each PDB has its shares increased to 50 from 1. They are then reallocated as needed from there.

The Resource Manager plan for CDBs manages resources at two levels:

- Assigning CPU shares among all the PDBs
- Prioritizing CPU access between Consumer Groups within each PDB

6.3.8.3 Choosing the Best Recommendation

Oracle Database Quality of Service Management can offer multiple recommendations for improving workload performance.

When trying to relieve a resource bottleneck for a particular Performance Class, Oracle Database QoS Management recommends adding more of the resource (such as CPU time) for that Performance Class or making the resource available more quickly to work requests in the Performance Class. The recommendations take the form of promoting the target Performance Class to a higher Consumer Group, demoting competing Performance Classes within the resource plan, adjusting CPU resources shared between different slices in a server pool, or moving servers between server pools.

Implementing a recommended action makes the resource less available to other Performance Classes. When generating recommendations, Oracle Database QoS Management evaluates the impact to system performance as a whole. If a possible recommendation for changing the allocation of resources provides a small improvement in the response time of one Performance Class, but results in a large decrease in the response time of another Performance Class, then Oracle Database

QoS Management reports that the performance gain is too small, and the change is not recommended.

Oracle Database QoS Management can issue recommendations that involve a negative impact to the performance of a Performance Class if:

- The negative impact on the Performance Class from which the resource is taken is projected not to cause a Performance Objective violation and a positive impact is projected for the Performance Class that gets better access to resources
- The Performance Class from which the resource is taken is lower ranked, and thus less critical to your business, than the Performance Class being helped

If the resource bottleneck can be resolved in multiple ways, then Oracle Database QoS Management recommends an action that is projected to improve the performance of the highest ranked Performance Class that is violating its objective. You can also view the alternative recommendations generated by Oracle Database QoS Management and see whether the action was recommended for implementation. For example, one possible solution to resolving a bottleneck on the CPU resource is to demote the Consumer Group associated with the Performance Class that is using the CPU the most. By limiting access to the CPU for the work requests in this Performance Class, the work requests in the other Performance Classes for that database get a larger share of the CPU time. However, Oracle Database QoS Management might decide not to recommend this action because the gain in response time for the target Performance Class is too small.

Related Topics

- [Viewing Recommendations](#)
- [Viewing Recommendation Details](#)

6.3.8.4 Contents of a Recommendation

Each recommendation is comprised of several pieces of information.

The analysis data for a recommendation include the projected change in response time for each Performance Class, the projected change in the [Performance Satisfaction Metric](#) (PSM) for each Performance Class, and the reason this action is chosen among other alternative actions, as shown in [Figure 6-7](#). In this example, if you implement the recommended action, then Oracle Database QoS Management predicts that the `sales cart` Performance Class, which has the highest ranking, will have an improvement in response time from 0.00510 seconds for database requests to 0.00426 seconds, which equates to an 11.6% gain in its PSM. The other Performance Classes are not affected by the change because they use a different server pool.

Figure 6-7 Example of the Analysis for a Recommended Action

Recommended Actions

Action Rank 1: Promote sales cart from Consumer Group 2 to Consumer Group 0.

Action Promote sales cart from Consumer Group 2 to Consumer Group 0.

Estimated Time 2 minutes

Rationale All potential single mapping changes have been analyzed. Changes evaluated and rejected are listed below.

Evaluation The beneficiary's PSM value is expected to change by 11.565 percentage points. The sum of all PSM values is expected to change by -27.265 percentage points. This action is a candidate for recommendation.

Projected Results	Performance Satisfaction Metric (Last 5 min)		Average Response Time			
	Performance Class	Projected (%)	Projected Change (%)	Objective Value (sec)	Current Value (sec)	Projected Value (sec)
	Default_pc	100	0.0	0.00000	0.00670	0.00959
hr_pc	68	0.0	0.00800	0.00259	0.00259	
sales_pc	-42	-38.8	0.00500	0.00514	0.00856	
erp_pc	74	0.0	0.01000	0.00262	0.00262	
sales cart	-30	11.6	0.00300	0.00510	0.00426	

Related Topics

- [Performance Satisfaction Metrics](#)
A useful metric for analyzing workload performance is a common and consistent numeric measure of how work requests in a Performance Class are doing against the current Performance Objective for that Performance Class.
- [Using Metrics to Identify Performance Issues](#)
Oracle Database QoS Management uses the metrics it collects to identify performance bottlenecks.
- [Viewing Recommendations](#)

6.3.8.5 Overview of Implementing Recommendations

Oracle Database QoS Management does not implement the recommendations automatically.

The recommended actions are performed only after the QoS Administrator clicks the **Implement** button. After the Oracle Database QoS Management administrator implements a recommendation, the system performance is reevaluated for the specified settling time before any new recommendations are made. You can also configure Enterprise Manager to generate alerts based upon the duration that a Performance Class has not been meeting its objective.

Related Topics

- [Using the Oracle Database QoS Management Dashboard](#)
The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.
- [Viewing Recommendations](#)
- [Implementing Recommendations](#)
By default, Oracle Database QoS Management does not automatically implement recommendations.
- [Reviewing Performance Metrics](#)
You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

6.3.8.6 Example: How Recommendations Are Generated

Consider a system that has two servers in an Online server pool, and two servers in a Back Office server pool. The Online server pool hosts two workloads: the `sales_pc` Performance Class and the `sales_cart` Performance Class. The minimum size of the Online server pool is two. The Back Office server pool hosts two internal applications: a human resources (HR) application and an enterprise resource planning (ERP) application. The Back Office server pool has a minimum size of one. The `sales_cart` Performance Class has the highest rank and the `erp_pc` Performance Class has the lowest rank. The `sales_pc` Performance Class is ranked higher than the `hr_pc` Performance Class.

In this scenario, if the `sales_pc` workload surges, causing contention for resources and causing the `sales_cart` Performance Class to violate its Performance Objective, then this could lead to a service-level agreement (SLA) violation for the OLTP application. Oracle Database QoS Management issues a recommendation to increase access to the CPU for the `sales_cart` Performance Class at the expense of the `sales_pc` workload, because the `sales_cart` Performance Class is of a higher rank; a higher rank indicates that satisfying the Performance Objective for the `sales_cart` Performance Class is more important than satisfying the Performance Objective for the `sales_pc` Performance Class.

If, after you implement the recommendation, the `sales_cart` and `sales_pc` Performance Classes are still not satisfying their Performance Objectives, then Oracle Database QoS Management issues a recommendation to increase the number of servers in the Online server pool by moving a server from the Back Office server pool, or a server pool that hosts less critical workloads or workloads with more headroom. In this scenario, a server can be moved from the Back Office server pool, because the Back Office server pool is currently above its minimum size of one. If the Back Office server pool had a minimum size of two, then Oracle Database QoS Management would have to find an available server in a different server pool; Oracle Database QoS Management does not recommend to move a server from a server pool if doing so will cause a server pool to drop below its minimum size.

If you implement the recommended action, and your applications use Cluster Managed Services and Client Run-time Load Balancing, then the application users should not see a service disruption due to this reallocation. The services are shut down transactionally on the server being moved. After the server has been added to the stressed server pool, all database instances and their offered services are started on the reallocated server. At this point, sessions start to gradually switch to using the new server in the server pool, relieving the bottleneck.

Using the same scenario, if the `sales_pc` Performance Class and `hr_pc` Performance Class both require additional servers to meet their Performance Objectives, then Oracle Database QoS Management first issues recommendations to improve the performance of the `sales_pc` Performance Class, because the `sales_pc` Performance Class is ranked higher than the `hr_pc` Performance Class. When the `sales_pc` Performance Class is satisfying its Performance Objectives, then Oracle Database QoS Management makes recommendations to improve the performance of the `hr_pc` Performance Class.

6.4 What Does Oracle Database QoS Management Manage?

Oracle Database QoS Management works with Oracle Real Application Clusters (Oracle RAC) and Oracle Clusterware. Oracle Database QoS Management operates over an entire Oracle RAC cluster, which can support a variety of applications.



Note:

Oracle Database QoS Management supports only OLTP workloads. The following types of workloads (or database requests) are not supported:

- Batch workloads
- Workloads that require more than one second to complete
- Workloads that use parallel data manipulation language (DML)
- Workloads that query GV\$ views at a signification utilization level

- [Managing Database Resources to Meet Service Levels](#)
- [High Availability Management and Oracle Database QoS Management](#)
Oracle Database QoS Management helps you achieve optimal performance levels for your application workloads.

6.4.1 Managing Database Resources to Meet Service Levels

Oracle Database QoS Management manages the CPU resource for a cluster. Oracle Database QoS Management does not manage I/O resources, so I/O intensive applications are not managed effectively by Oracle Database QoS Management.

Oracle Database QoS Management integrates with the Oracle RAC database through the following technologies to manage resources within a cluster:

- Database Services
- Oracle Database Resource Manager
- Oracle Clusterware
- Run-time Connection Load Balancing

Oracle Database QoS Management periodically evaluates the resource wait times for all used resources. If the average response time for the work requests in a Performance Class is greater than the value specified in its Performance Objective,

then Oracle Database QoS Management uses the collected metrics to find the bottlenecked resource. If possible, Oracle Database QoS Management provides recommendations for adjusting the size of the server pools or making alterations to the consumer group mappings in the resource plan used by Oracle Database Resource Manager.

- [Database Services](#)
Database services provide a mechanism you can use to group together related work requests.
- [Oracle Database Resource Manager](#)
Oracle Database QoS Management uses Oracle Database Resource Manager to manage allocate resources to performance classes.
- [Oracle Clusterware](#)
Oracle Database QoS Management manages and monitors server pools configured by Oracle Clusterware.
- [Run-time Connection Load Balancing](#)

6.4.1.1 Database Services

Database services provide a mechanism you can use to group together related work requests.

An application connects to the cluster databases using database services. A user-initiated query against the database could use a different service than a web-based application. Different services can represent different types of work requests. Each call or request made to the Oracle RAC database is a *work request*.

You can also use database services to manage and measure database workloads for policy-managed, administrator-managed, and multitenant databases. To manage the resources used by a service, some services might be deployed on several Oracle RAC instances concurrently, whereas others might be deployed on only a single instance to isolate the workload that uses that service.

In an Oracle RAC cluster, Oracle Database QoS Management monitors the server pools and its nodes, on which the database services are offered. Services are created by the database administrator for a database. For a policy-managed database, the service runs on all servers in the specified server pool. If a singleton service is required due to the inability of the application to scale horizontally, then the service can be restricted to run in a server pool that has a minimum and maximum size of one. Policy-managed singleton service support in pools larger than one is restricted to measurement and monitoring only.

To use Oracle Database QoS Management for managing performance, create one or more policy-managed databases that run in server pools. If you have administrator-managed databases, then the database instances are placed in the Generic server pool and Oracle Database QoS Management can only monitor these databases.

When you first configure Oracle Database QoS Management, a default Performance Policy is created for each service that is discovered on the server pools being monitored. The name of these default Performance Classes are *service_name_pc*. The workload you want to monitor and manage the resource for must use a database service to connect to the database.

Related Topics

- [Oracle Real Application Clusters Administration and Deployment Guide](#)

6.4.1.2 Oracle Database Resource Manager

Oracle Database QoS Management uses Oracle Database Resource Manager to manage allocate resources to performance classes.

Oracle Database Resource Manager (Resource Manager) is an example of a resource allocation method; Resource Manager can allocate CPU shares among a collection of resource consumer groups based on a resource plan specified by an administrator. A resource plan allocates the percentage of opportunities to run on the CPU.

Oracle Database QoS Management does not adjust existing Resource Manager plans; Oracle Database QoS Management activates a resource plan named `APPQOS_PLAN`, which is a complex, multilevel resource plan. Oracle Database QoS Management also creates consumer groups that represent Performance Classes and resource plan directives for each consumer group.

When you implement an Oracle Database QoS Management recommendation to promote or demote a consumer group for a Performance Class, Oracle Database QoS Management makes the recommended changes to the mapping of the Performance Class to the CPU shares specified in the resource plan. By altering the consumer group, the Performance Class that is currently not meeting its Performance Objective is given more access to the CPU resource.

For multitenant databases, instead of managing CPU shares directly for a PDB, Oracle Database QoS Management manages the CPU shares for the multitenant database by assigning PDB shares to each PDB in the CDB. Initially, each PDB in a multitenant database is assigned 50 PDB shares. If a Performance Class is not meeting its performance objectives, then PDB shares are reassigned from a donor PDB and assigned to the target PDB. Oracle Database QoS Management manages the assignment of PDB shares in a resource plan. The two resource plans used by Oracle Database QoS Management for multitenant databases are:

- `ORA$QOS_CDB_PLAN`: governs CPU shares for each PDB
- `ORA$QOS_PLAN`: governs the PDB shares



Note:

Do not edit the Oracle Database QoS Management resource plans except as specified in "[Editing the Resource Plan for Oracle Database QoS Management](#)".

Related Topics

- [Enabling Oracle Database QoS Management](#)
If you have multiple databases running on the same cluster, you can specify which databases are managed by Oracle QoS Management.
- *Oracle Database Administrator's Guide*

6.4.1.3 Oracle Clusterware

Oracle Database QoS Management manages and monitors server pools configured by Oracle Clusterware.

You must have Oracle Clusterware installed and configured before you can use Oracle Database QoS Management. The cluster administrator should create server pools to be used to deploy policy-managed Oracle RAC databases. Administrator-managed Oracle RAC databases use only the Generic server pool.

When you first configure Oracle Database QoS Management and create the initial Policy Set, you specify which server pools should be managed by Oracle Database QoS Management and which should only be monitored. If you select a server pool to be managed by Oracle Database QoS Management, then Oracle Database QoS Management monitors the resources used by all the Performance Classes that run in that server pool. If a Performance Class is not satisfying its Performance Objective, then Oracle Database QoS Management can recommend moving servers between server pools to provide additional resources where needed.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

6.4.1.4 Run-time Connection Load Balancing

Run-time connection load balancing enables Oracle Clients to provide intelligent allocations of connections in the connection pool when applications request a connection to complete some work; the decision of which instance to route a new connection to is based on the current level of performance provided by the database instances.

Applications that use resources managed by Oracle Database QoS Management can also benefit from connection load balancing and transparent application failover (TAF). Connection load balancing enables you to spread user connections across all of the instances that are supporting a service. For each service, you can define the method you want the listener to use for load balancing by setting the connection load balancing goal, using the appropriate SRVCTL command with the `-clbgoal` option. You can also specify a single TAF policy for all users of a service using SRVCTL with the options `-failovermethod,-failovertime`, and so on.

See Also:

- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about run-time connection load balancing
- ["Supported Service Configurations"](#)
- *Oracle Database Net Services Administrator's Guide* for more information about configuring TAF

6.4.2 High Availability Management and Oracle Database QoS Management

Oracle Database QoS Management helps you achieve optimal performance levels for your application workloads.

Performance management and managing systems for high availability are closely related. Users typically consider a system to be up, or available, only when its

performance is acceptable. You can use Oracle Database QoS Management and Performance Objectives to specify and maintain acceptable performance levels.

Oracle Database QoS Management is a run-time performance management product that optimizes resource allocations to help your system meet service-level agreements under dynamic workload conditions. Oracle Database QoS Management provides recommendations to help the work that is most critical to your business get the necessary resources. Oracle Database QoS Management assists in rebalancing resource allocations based upon current demand and resource availability. Nonessential work is suppressed to ensure that work vital to your business completes successfully.

Oracle Database QoS Management is not a feature to use for improving performance; the goal of Oracle Database QoS Management is to maintain optimal performance levels. Oracle Database QoS Management assumes that system parameters that affect both performance and availability have been set appropriately, and that they are constant. For example, the `FAST_START_MTTR_TARGET` database parameter controls how frequently the database writer checkpoints blocks to the data files to minimize instance recovery time. Using a low value for this parameter reduces the amount of time required to recover your database, but the overhead of writing redo log data more frequently can have a negative impact on the performance of your database. Oracle Database QoS Management does not make recommendations regarding the values specified for such parameters.

Management for high availability encompasses many issues that are not related to workload and that cannot be affected by managing workloads. For example, system availability depends crucially on the frequency and duration of software upgrade events. System availability also depends directly on the frequency of hardware failures. Managing workloads cannot change how often software upgrades are done or how often hardware fails.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*
- *Oracle Database Performance Tuning Guide*
- *Oracle Database High Availability Overview*

6.5 Overview of Metrics

Oracle Database QoS Management uses a standardized set of metrics, which are collected by all the servers in the system.

Oracle Database QoS Management bases its decisions on observations of how long work requests spend waiting for resources. Examples of resources that work requests can wait for include hardware resources, such as CPU cycles, disk I/O queues, and Global Cache blocks. Other waits can occur within the database, such as latches, locks, pins, and so on. Although the resource waits within the database are accounted for in the Oracle Database QoS Management metrics, they are not managed or specified by type.

The response time of a work request consists of execution time and a variety of wait times; changing or improving the execution time generally requires application source code changes. Oracle Database QoS Management therefore observes and manages only wait times.

There are two types of metrics used to measure the response time of work requests: performance metrics and resource metrics. These metrics enable direct observation of the wait time incurred by work requests in each Performance Class, for each resource requested, as the work request traverses the servers, networks, and storage devices that form the system. Another type of metric, the Performance Satisfaction Metric, measures how well the Performance Objectives for a Performance Class are being met.

- [Performance Metrics](#)
Performance metrics give an overview of where time is spent in the system and enable comparisons of wait times across the system.
- [Resource Metrics](#)
Resource metrics give an overview of where time is spent in the system for each resource requested.
- [Performance Satisfaction Metrics](#)
A useful metric for analyzing workload performance is a common and consistent numeric measure of how work requests in a Performance Class are doing against the current Performance Objective for that Performance Class.
- [Using Metrics to Identify Performance Issues](#)
Oracle Database QoS Management uses the metrics it collects to identify performance bottlenecks.

Related Topics

- [Using the Oracle Database QoS Management Dashboard](#)
The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.

6.5.1 Performance Metrics

Performance metrics give an overview of where time is spent in the system and enable comparisons of wait times across the system.

Performance metrics are collected at the entry point to each server in the system. Data is collected periodically and forwarded to a central point for analysis, decision making, and historical storage. See [Figure 6-6](#) for an illustration of how the system data are collected.

Performance metrics measure the response time (the difference between the time a request comes in and the time a response is sent out). The response time for all database client requests in a Performance Class is averaged and presented as the average response time, measured in seconds for a database request.

Related Topics

- [Reviewing Performance Metrics](#)
You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

6.5.2 Resource Metrics

Resource metrics give an overview of where time is spent in the system for each resource requested.

There are two resource metrics for each resource of interest in the system:

- Resource usage time—measures how much time was spent using the resource for each work request
- Resource wait time—measures the time spent waiting to get the resource

Resources are classified as CPU, Storage I/O, Global Cache, and Other (database waits). The data are collected from the Oracle RAC databases, Oracle Clusterware, and the operating system.

Related Topics

- [Viewing the Resource Wait Times Breakdown](#)

6.5.3 Performance Satisfaction Metrics

A useful metric for analyzing workload performance is a common and consistent numeric measure of how work requests in a Performance Class are doing against the current Performance Objective for that Performance Class.

This numeric measure is called the Performance Satisfaction Metric.

Different performance objectives are used to measure the performance of workloads, as shown in the following table:

Workload Type	Performance Objectives
OLTP	Response time, transactions per second
Batch	Velocity, throughput
DSS	Read or cache hit ratio, duration, throughput

Oracle Database QoS Management currently supports only OLTP workloads. For OLTP workloads, you can only configure a response time performance objective.

Related Topics

- [Interpreting the Performance Overview Graphs](#)

6.5.4 Using Metrics to Identify Performance Issues

Oracle Database QoS Management uses the metrics it collects to identify performance bottlenecks.

The Oracle Database QoS Management metrics provide the information needed to systematically identify Performance Class bottlenecks in the system. When a Performance Class is violating its Performance Objective, the bottleneck for that Performance Class is the resource that contributes the largest average wait time for each work request in that Performance Class.

The Oracle Database QoS Management metrics are used to find a bottleneck for a Performance Class using the following steps:

1. Oracle Database QoS Management selects the highest ranked Performance Class that is not meeting its Performance Objective.
2. For that Performance Class, wait times for each resource are determined from the collected metrics.
3. The resource with the highest wait time per request is determined to be the bottlenecked resource.

Analyzing the average wait for each database request and the total number of requests for each Performance Class provides the resource wait time component of the response times of each Performance Class. The largest such resource contribution (CPU, Storage I/O, Global Cache, or Other) is the current bottleneck for the Performance Class.

Related Topics

- [Viewing the Resource Wait Times Breakdown](#)

7

Supported Workloads and Strategies

This chapter discusses the different ways that you can use Oracle Database Quality of Service Management (Oracle Database QoS Management) to manage the workload on your system and the best practices for configuring your system to effectively use Oracle Database QoS Management.

- [Supported Configurations for Oracle Database QoS Management](#)
To use Oracle Database QoS Management, your system must meet certain requirements. Also, your applications and database connections must conform to certain standards.
- [Strategies for Creating Classifiers for Performance Classes](#)
Review the requirements for performance classes and workload classifiers.
- [Configuration Strategies for Effective Resource Management](#)
This section discusses key configuration recommendations and requirements for systems that are managed by Oracle Database QoS Management.
- [Sample Implementation of Oracle Database QoS Management](#)
This section describes a sample implementation of Oracle Database QoS Management. The process by which Oracle Database QoS Management manages performance is described.
- [Creating Oracle Database QoS Management Performance Policies for the Demo System](#)
- [Managing Service Levels with Oracle Database QoS Management](#)

7.1 Supported Configurations for Oracle Database QoS Management

To use Oracle Database QoS Management, your system must meet certain requirements. Also, your applications and database connections must conform to certain standards.

If you do not use a supported configuration, then Oracle Database QoS Management reports a configuration violation and is disabled.

- [Supported Server Pool Configurations](#)
- [Supported Database Configurations](#)
Depending on your database configuration, all or only some of the Oracle Database QoS Management features are available,
- [Supported Service Configurations](#)
- [Supported Workload and Objective Types](#)
Review the types of database workloads and objectives that are supported with Oracle Database QoS Management.

7.1.1 Supported Server Pool Configurations

Before Oracle Database Quality of Service (QoS) Management can manage the resources for your cluster, you must mark server pools as managed in your Performance Policy. If a server pool is not managed by Oracle Database QoS Management, then the server pool is not visible to the Oracle Database QoS Management server. Any workloads or resources in the unmarked server pools are not managed by Oracle Database QoS Management.

You can select server pools to be managed by Oracle Database QoS Management even if they do not contain any database instances, or have a current size of zero. This enables you to preconfigure a server pool for management by Oracle Database QoS Management before the workload is started. Also, if you configure a server pool to have a minimum size of zero, then Oracle Database QoS Management can remove the servers from that server pool to provide resources for higher priority workloads. In this case, even though there are no resources within the server pool for Oracle Database QoS Management to manage, any workloads that run in that server pool are still monitored and, if possible, resources are allocated to that server pool to support the workloads.

7.1.2 Supported Database Configurations

Depending on your database configuration, all or only some of the Oracle Database QoS Management features are available,

All the features of Oracle Database QoS Management are available for Oracle RAC databases running Oracle Database 12c release 2 software. For Oracle Database 11g release 2 and Oracle Database 12c release 1 Oracle RAC databases, your Oracle RAC database must be policy-managed database, or you can only use Oracle Database QoS Management in Measure-only or Monitor (12.1.0.2) modes.

For policy-managed databases, the database services should be created as `UNIFORM` services, meaning the service is offered by every available instance running in the specified server pool. If your application requires a `SINGLETON` service, then, to use Oracle Database QoS Management in management mode, the service must run in a server pool that has a maximum size of one. If you use a `SINGLETON` service in a server pool that has a maximum size greater than one, then Oracle Database QoS Management reports a configuration violation. If you selected Measure-only for the management mode for this database, then you can use `SINGLETON` services any configuration.

Oracle Database QoS Management supports multiple databases sharing a server pool. If you have multiple databases using the same server pool, then every database that uses the server pool must have Oracle Database QoS Management enabled if they all have performance classes defined. Oracle Database QoS Management also supports Oracle RAC One Node databases (sometimes referred to as *singleton databases*), but these databases must use server pools that have a maximum size of one if their performance classes are being managed and not simply measured or monitored.

When you create a database, the default value of the `CPU_COUNT` initialization parameter for the database instance is set to the value of the number of physical CPUs on each node that the instance runs on. If you have multiple database instances on the same node and you have performance classes in Management mode, then you must adjust the value of `CPU_COUNT` for each instance so that the sum of `CPU_COUNT` for each instance

that runs on the node is equal to or less than the physical number of CPUs on that node. Also, the value of `CPU_COUNT` must be the same for every instance of a database. For example, for the `sales` database, you cannot have `CPU_COUNT` set to four for the `sales1` instance and `CPU_COUNT` set to two for the `sales2` instance if both instances are in the same server pool.

Starting with Oracle Database 12c release 2 (12.2), Oracle Database QoS Management provides full support for multitenant databases. In the multitenant architecture, an Oracle database functions as a multitenant container database (CDB). A CDB includes zero, one, or many customer-created pluggable databases (PDBs). A PDB is a portable collection of schemas, schema objects, and nonschema objects that appears to an Oracle Net client as a non-CDB. All Oracle databases before Oracle Database 12c were non-CDBs.

The Oracle Multitenant option enables you to consolidate data and code without altering existing schemas or applications. A PDB behaves the same as a non-CDB as seen from a client connecting with Oracle Net. Operations that act on an entire non-CDB act in the same way on an entire CDB, for example, when using Oracle Data Guard and database backup and recovery. Thus, the users, administrators, and developers of a non-CDB have substantially the same experience after the database has been consolidated. By consolidating hardware and sharing database memory and files, you reduce costs for hardware, storage, availability, and labor. For example, 100 PDBs on a single server can share one database instance and one set of database files, thereby requiring less hardware and fewer personnel.

Oracle Database QoS Management supports:

- Schema consolidation within a pluggable database (PDB) by adjusting the CPU shares of Performance Classes running in the PDB
- Database consolidation by adjusting the CPU shares between different PDBs within the same multitenant container database (CDB)
- Multiple CDB consolidation by adjusting CPU counts for all CDBs hosted on the same physical servers and the total number of CDB instances by varying the server pool size (if using policy-managed databases)

This functionality is seamlessly integrated into the Oracle Database QoS Management pages in Oracle Enterprise Manager Cloud Control.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

7.1.3 Supported Service Configurations

Database services that are managed by Oracle Clusterware are required for Oracle Database QoS Management. All workloads managed by Oracle Database QoS Management must connect to the database using a database service that is managed by Oracle Clusterware. You cannot use the default database service; the default database service is not managed by Oracle Clusterware.

The services used to connect to the database should be `UNIFORM`. If your application requires a `SINGLETON` service, then, to use Oracle Database QoS Management, the service must run in a server pool that has a maximum size of one.

The Oracle RAC high availability framework monitors the database and its services and sends event notifications using fast application notification (FAN). Oracle

Clusterware and Oracle Net Services provide load balancing for services according to rules specified in the service configuration. These rules are:

- The **connection load balancing goal**: Connections are routed to an instance using the current workload for that instance and the type of connection (`LONG` or `SHORT`) to determine which instance can provide the best performance. For Oracle Database QoS Management, the connection load balancing goal should be set to `LONG`, so that if a new server is allocated to a server pool for a service, new connections will migrate faster to the new server. By migrating connections more quickly to a new server, the workload is balanced faster across all the available instances and the response times for workloads improve at a faster rate.
- The **run-time connection load balancing goal**: The load balancing advisory data is used to determine which instance best meets the goal specified for the service. The two goals are `SERVICE_TIME`, for which load balancing advisory data is based on elapsed time for work done in the instance, and `THROUGHPUT`, for which the load balancing advisory data is based on the rate that work is completed in the instance. For Oracle Database QoS Management, the run-time connection load balancing goal should be set to `SERVICE_TIME` for all database services that use server pools except optionally those with a maximum size of one.

Run-time connection load balancing sends advice to connection pools on how to balance connection requests across instances in an Oracle RAC database. The load balancing advisory also provides advice about how to direct incoming work to the instances that provide the optimal quality of service for that work. This minimizes the need to relocate the work later.

To configure the load balancing goals for a service, use the Server Control (SRVCTL) utility, as shown in the following example, or use Enterprise Manager:

```
srvctl modify service -db db_name -service service_name -rlbgoal SERVICE_TIME -  
clbgoal LONG
```

7.1.4 Supported Workload and Objective Types

Review the types of database workloads and objectives that are supported with Oracle Database QoS Management.

In the initial release of Oracle Database QoS Management, only online transaction processing (OLTP) workloads are supported. The only supported Performance Objective is average response time for database requests. Oracle Database QoS Management is designed to manage open workloads, or a workload in which demand is independent of the response time.

The database requests for your application workload must have an average response time of less than one second, and preferably, an average response time of less than 0.5 seconds. Each database request within a Performance Class should be homogenous with respect to resource usage. If a subset of the database requests in a workload use significantly more resources than the other requests, then you should create a new Performance Class to contain the database requests that require more resources.

Oracle Database QoS Management does not support workloads that involve parallel queries. By default, parallel queries run on all available instances of the database, regardless of which service was used to connect to the database; the workload is not contained, or restrained to running on only those instances that offer the service. For a similar reason, Oracle Database QoS Management does not support workloads that involve a significant amount of database requests involving queries to GV\$ views.

For a workload to be managed by Oracle Database QoS Management, the database connections must use a database service that is managed by Oracle Clusterware. The client or application that initiates the connection must be a JDBC (thick or thin) client, or an OCI client. Workloads that use a bequeath connection to the database are not managed by Oracle Database QoS Management.

Related Topics

- [Deciding to Create New Performance Classes](#)
Over time, your workload and performance goals can change.

7.2 Strategies for Creating Classifiers for Performance Classes

Review the requirements for performance classes and workload classifiers.

Currently, the only workloads that Oracle Database QoS Management manages are OLTP database workloads. To manage the workload for a database, the incoming work requests must be assigned to a Performance Class. Workload is mapped to a Performance Class using classifiers.

In multi-tier environments, a request from a client is routed to different database sessions by the middle tier or through load balancing, making the tracking of a client across database sessions difficult. Classifiers use session attributes to identify work requests. The attributes used are service name, user name, module, action, and program.

Each classifier must specify one or more service names. If a classifier specifies multiple service names, then when matching the connection data to a Performance Class, the service names are evaluated using an `OR` operation. If any one of the service names specified in the classifier matches the service name in the work request, then the comparison evaluates to `TRUE`.

To set the `MODULE` and `ACTION` attributes, use the `OCIAttrSet()` call. Use the default namespace, `USERENV`, for the application context.

You can also optionally include the **UserName** and **program name** in the classifier. The user name is the name of the database user to which the session connects. The program attribute is the name of the client program used to log in to the server.

If the classifier for a Performance Class specifies multiple attributes, then the session attributes are combined using an `AND` operation. If all of the attribute values specified in the classifier match the session attribute values in the work request, then the comparison evaluates to `TRUE`. If you have more than one classifier that uses similar attribute values, then place the classifier with the most fine-grained conditions first.

For example, consider the following classifiers:

- `create_invoice_taxes_pc` that specifies the `sales_cart` service, the `ORDER` module, and the `CALCULATE TAX` action
- `create_invoice_pc`, which specifies the `sales_cart` service and the `ORDER` module

The `create_invoice_taxes_pc` classifier should be evaluated before the `create_invoice_pc` classifier. If a work request uses the `sales_cart` service, and is performing the `CALCULATE TAX` action in the `ORDER` module, then the work request is assigned to the `create_invoice_taxes_pc`. If the work request does not have the matching values for all the attributes, then the work request is compared to the next

classifier, for `create_invoice_pc`. If you evaluate the `create_invoice_pc` classifier first, then any work request that uses the `sales_cart` service and the `ORDER` module will be assigned to the `create_invoice_pc` Performance Class, regardless of what action the work request performs.

You can create up to 47 Performance Classes for a cluster. If you have more than 47 services for your cluster, then use more than one service name within classifiers. Once a match is found for a classifier, Oracle Database QoS Management assigns a tag to the matching work request. The tag is based on the classifier that evaluates to `TRUE`.

Related Topics

- [Using Additional Filters for Classifying Work Requests](#)
- [Applying Classifiers to Work Requests](#)
- [Performance Class Tags](#)

7.3 Configuration Strategies for Effective Resource Management

This section discusses key configuration recommendations and requirements for systems that are managed by Oracle Database QoS Management.

- [About Resource Bottlenecks](#)
Oracle Database QoS Management measures use and wait times for CPU, Global Cache, I/O, and other resources to determine where a bottleneck is located.
- [CPU Resource Bottlenecks](#)
- [Configuration Recommendations for Global Cache Resource Bottlenecks](#)
- [Configuration Recommendations for I/O Resource Bottlenecks](#)
- [Configuration Recommendations for Other Types of Bottlenecks](#)

7.3.1 About Resource Bottlenecks

Oracle Database QoS Management measures use and wait times for CPU, Global Cache, I/O, and other resources to determine where a bottleneck is located.

The target Performance Class and its bottle-necked resource are identified on the Oracle Database QoS Management Dashboard (the Dashboard), however, only the CPU resource is actively managed in this release.

7.3.2 CPU Resource Bottlenecks

A CPU resource bottleneck is detected when there are excessive wait times on the collection of CPU queues running that workload. Oracle Database QoS Management offers recommendations you can implement to relieve the bottleneck.

One solution to this type of bottleneck is to increase the number of opportunities for the workload to run on the CPU. Oracle Database QoS Management implements this solution by assigning the workload to a consumer group that has more CPU shares across the server pool.

Another solution is to provide more CPU resources. If you have multiple instances sharing the CPU resources for each server in the server pool, and you have

implemented instance caging, then Oracle Database QoS Management can suggest altering the CPU counts for the instances in the server pool; this solution gives more CPU resources to the workloads that are not meeting performance expectations by taking CPU resources away from an instance that is of lower rank or has the headroom to contribute the resources.

If there is a CPU resource bottleneck that cannot be relieved by adjusting the CPU counts between instances, then Oracle Database QoS Management can recommend moving a new server into the server pool. The server can come from the Free pool, from a less-stressed server pool, or from a server pool that hosts a less critical workload.

7.3.3 Configuration Recommendations for Global Cache Resource Bottlenecks

A Global Cache resource bottleneck is detected when there is excessive data block movement between database instances. This is usually caused by an application that is not configured properly or is not able to scale horizontally. Configuring the application to run in a server pool with a maximum size of one or partitioning the data can usually relieve the bottleneck.

Oracle Database QoS Management cannot perform either of these actions in this release and does not provide a recommendation that can be implemented for this type of bottleneck.

7.3.4 Configuration Recommendations for I/O Resource Bottlenecks

An I/O resource bottleneck is detected when there are excessive wait times on the storage subsystem. This type of bottleneck is typically caused by either too few disk spindles or not enough network bandwidth on the storage interconnect. To resolve this bottleneck, spread the database files across a higher number of disks, or configure a separate network interface card (NIC) for a dedicated storage interconnect.

Oracle Database QoS Management cannot resolve this type of bottleneck in this release and does not provide a recommendation that can be implemented.

7.3.5 Configuration Recommendations for Other Types of Bottlenecks

The last resource type used to categorize bottlenecks, Other, is used for all other wait times. These database wait times are usually caused by SQL performance issues that result from an application that is not optimized, waiting on latches, and so on. These bottlenecks can be investigated using Oracle Database tuning tools such as Oracle Cluster Health Advisor, Automatic Workload Repository (AWR) and Automatic Database Diagnostic Monitor (ADDM).

Resolving these types of bottlenecks are outside the scope of the run-time system management provided by Oracle Database QoS Management and Oracle Database QoS Management does not provide any recommendations that can be implemented

7.4 Sample Implementation of Oracle Database QoS Management

This section describes a sample implementation of Oracle Database QoS Management. The process by which Oracle Database QoS Management manages performance is described.

- [Description of the Demo System](#)
The sample implementation uses a four-node cluster running on Linux.
- [Description of the System Workload](#)
There are many different types of workloads for a database.
- [Initial Oracle Database QoS Management Configuration](#)
You must configure Oracle Database QoS Management before you can use it.

7.4.1 Description of the Demo System

The sample implementation uses a four-node cluster running on Linux.

The nodes are named `test_rac1` to `test_rac4`. In normal operation, each node does the following:

Node	Purpose	Services
<code>test_rac1</code>	Runs Oracle Grid Infrastructure for a cluster and the first database instance for the <code>backoffice</code> database	HR and ERP
<code>test_rac2</code>	Runs Oracle Grid Infrastructure for a cluster and the second database instance for the <code>backoffice</code> database	HR and ERP
<code>test_rac3</code>	Runs Oracle Grid Infrastructure for a cluster and the first database instance for the <code>online</code> database	Sales and Sales_Cart
<code>test_rac4</code>	Runs Oracle Grid Infrastructure for a cluster and the second database instance for the <code>online</code> database	Sales and Sales_Cart

The cluster is logically divided into two server pools with the following constraints:

Name	Min Size	Max Size	Current Size	Importance
<code>backoffice</code>	1	-1	2	1
<code>online</code>	1	-1	2	2
Free	0	-1	0	0

The server pool constraints as shown here guarantee that at least one server is allocated to each of the server pools (and the databases that run in those server pools) and the remaining servers can be shared on a demand basis to manage service levels. The `online` server pool hosts the most critical workloads, because it has the highest value for Importance. If a server failure occurs, then maintaining the minimum size for the `online` server pool takes priority over maintaining the minimum size of the other server pools.

7.4.2 Description of the System Workload

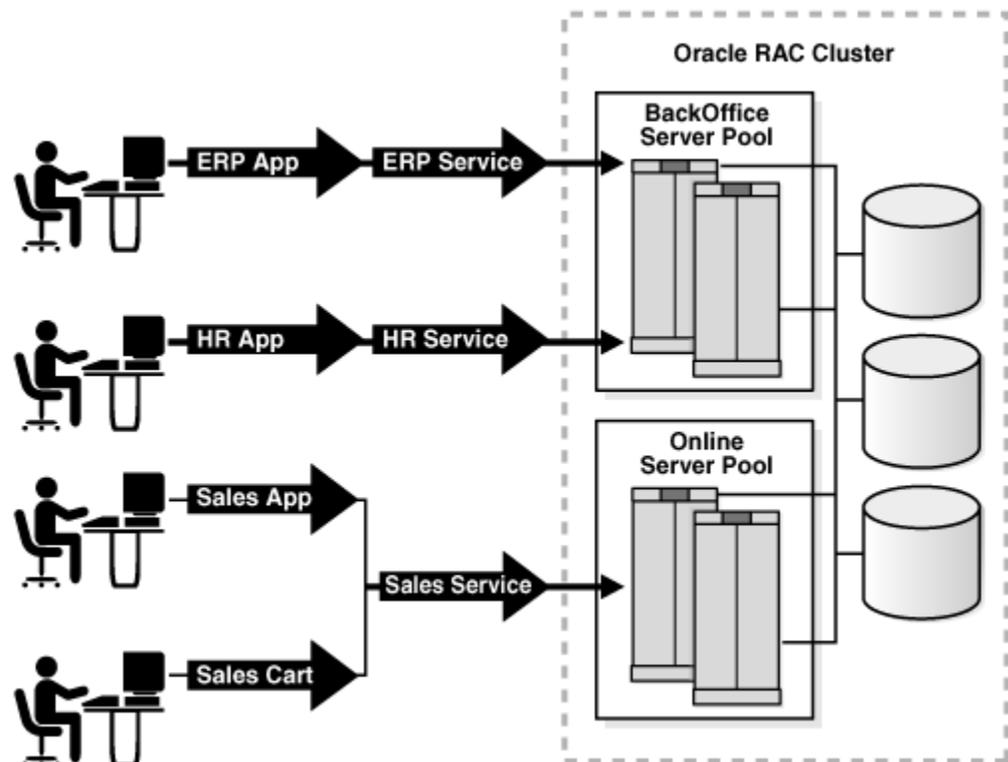
There are many different types of workloads for a database.

This release of Oracle Database QoS Management focuses on managing OLTP workloads, which are the type most likely to have an **open workload** (workloads for which demand remains constant even as system performance degrades) and be vulnerable to outages due to workload surges. For this demonstration, assume there is a combination of internal and external workloads hosted in the same cluster so the resources can be shared.

There are four types of workloads demonstrated for this demo system, as illustrated in [Figure 7-1](#):

- An ERP application based on J2EE that connects to the database instances in the `backoffice` server pool using the `ERP` service
- An internal HR application based on Oracle C Interface (OCI) that connects to the database instances in the `backoffice` server pool using the `HR` service
- An external Sales application based on J2EE that connects to the database instances in the `online` server pool using the `Sales` service
- An external Sales checkout application (Sales Cart) based on J2EE that connects to database instances through a specific database user in the `online` server pool using the `Sales` service

Figure 7-1 Illustration of a Sample Workload



By using two server pools, the workloads and their dependent databases are logically separated but can readily share resources between them.

7.4.3 Initial Oracle Database QoS Management Configuration

You must configure Oracle Database QoS Management before you can use it.

At first, there is no Oracle Database QoS Management configured for this system. Using Oracle Enterprise Manager Cloud Control, there are two configuration workflows to complete to enable Oracle Database QoS Management for the cluster. The first workflow configures each database for Oracle Database QoS Management and the second workflow configures and enables Oracle Database QoS Management for the cluster.

After you create a default Policy Set, using the database services that are discovered automatically, Oracle Database QoS Management can be fine-tuned to align the workloads with their respective service-level agreements or objectives.

Related Topics

- [Enabling Oracle Database QoS Management](#)
If you have multiple databases running on the same cluster, you can specify which databases are managed by Oracle QoS Management.

7.5 Creating Oracle Database QoS Management Performance Policies for the Demo System

In this section, the sample implementation of Oracle Database QoS Management is further evolved to include creating and activating Performance Policies and refining them with additional Performance Classes.

Because the default Performance Policy is created by discovering the database services in measure-only mode, the default Performance Policy can initially be activated to test how all of the workloads perform in the cluster. The Dashboard displays both the resource use and wait times that comprise the average response time for each Performance Class during different periods of demand. These numbers can serve to help understand the minimum response times achievable with the allocated resources

If your workloads peak at different times on a regular basis or your service-level agreements (SLAs) are variable based upon time, day of week, and so on, then create additional measure-only Performance Policies that change the size of the server pools to evaluate the minimum resources required for your workloads. In this demonstration, for the Sales application, the workload that uses the `online` server pool requires a minimum of two servers. The `backoffice` server pool requires only one server to satisfy the workload requests. If both server pools currently contain two servers, then you can enable the `online` server pool to take a server from the `backoffice` server pool, if needed, by setting the minimum size of the `backoffice` server pool to one. You would use a server pool directive override in the "Business Hours" Performance Policy to specify the minimum size of one for the `backoffice` server pool.

You could interpret the minimum size of a server pool as the number of servers *owned* by that server pool. If the sum of the minimum sizes of all the server pools in the cluster is less than the number of servers in the cluster, then the extra servers are referred to as *floaters*, which are shared by all server pools. For example, if your

cluster has 15 servers, three server pools, and a minimum size of four for each server pool, then your system has three floaters.

After the Performance Policies have been run in measure-only mode, Performance Objectives can be added to each Performance Class. The Performance Objectives can be ranked based upon how critical the maintenance of that Performance Objective is to your business. Performance Objectives should be set to maximize [headroom](#) above the observed response times but below the response times required to meet SLAs. Maintaining at least 50% headroom is a good starting point to support trading off resources should a Performance Class experience a workload surge. For example, if a Performance Class has an average response time of two milliseconds (ms), then the Performance Objective could be set to three ms— two ms response time and an additional one ms which corresponds to the 50% headroom.

Although service-based classifiers can provide for easy configuration, you might want to define more than one Performance Objective for a service. For example, the `sales` service can contain many different workloads, such as Browse Products, Add Customer Account, Sales Cart and Browse Orders. Because the Sales Cart workload generates revenue, you might want this workload to have a shorter response time than the other workloads. You must create a separate Performance Class and associated classifiers to specify specific Performance Objectives for the different workloads.

On the Define Classifier page in the Policy Set wizard, a sales cart performance classifier can be defined by entering `sales` as the Service Name and if the application can set `MODULE` or `ACTION`, enter an appropriate value, otherwise configure a separate `USERNAME` from the middle tier. As soon as this new Performance Class is defined, the Performance Class appears automatically in all of the Performance Policies in measure-only mode. The new Performance Class is given the lowest rank by default. Use these values initially to test the performance of your system. After the average performance levels can be determined, a Performance Objective and rank for this Performance Class can be set within each Performance Policy.

7.6 Managing Service Levels with Oracle Database QoS Management

The implementation of Oracle Database QoS Management is completed by actively managing the service levels, which means responding to alerts, reviewing and implementing recommendations, and tracking results. This section describes the actions you would perform on the demo system.

After all the workloads run and the Dashboard displays the performance of the demo system, you need to be alerted should a workload surge or failure cause a Performance Objective to stop being met. The Performance Satisfaction Metric (PSM) normalizes all of the objectives and provides a quick way to observe the health of the system. By observing the PSM Trend indicator you can see how well a Performance Class is meeting its objective over the last five minutes, and problems can be observed. Performance Objective violations produce recommendations that state how resources should be reallocated to relieve the bottleneck. Details and projections are available for further analysis of the bottleneck and possible solutions. If the recommendation is an action that can be implemented by Oracle Database QoS Management, then an **Implement** button is displayed.

Performance Objective violations of short duration are tolerated in most SLAs. Therefore, Enterprise Manager alerts can be configured by Performance Class

specifying the duration of continuous violation. These alerts are configured on the Database alert page, but can be defined for all Performance Classes in the cluster.

An audit log of policy changes, violations and actions is available in the Oracle Grid Infrastructure home in the `$ORACLE_BASE/crsdata/host/qos/logs/dbwlm/auditing` directory on the server that hosts the Oracle Database QoS Management server. To determine which server is hosting the Oracle Database QoS Management server, enter the following command at the operating system prompt:

```
srvctl status qosmsserver
```

8

Installing and Enabling Oracle Database QoS Management

This chapter describes the tasks you must complete to install and configure Oracle Database QoS Management on your system. Some of the tasks in this section must be performed by the cluster administrator.

- [Configuring Oracle Database QoS Management to Manage Oracle Database Workloads](#)

Before you can use Oracle Database Quality of Service (QoS) Management, you must configure the databases.

See Also:

Oracle Clusterware Administration and Deployment Guide for information about the cluster administrator

8.1 Configuring Oracle Database QoS Management to Manage Oracle Database Workloads

Before you can use Oracle Database Quality of Service (QoS) Management, you must configure the databases.

When configuring the Oracle RAC databases to work with Oracle Database QoS Management, the server pool configuration tasks are not required for administrator-managed databases.

1. For policy-managed databases, the database administrator (DBA) requests access to a server pool to be used for the database. The cluster administrator creates a server pool for the DBA and grants access to this server pool to the DBA. If the cluster administrator user is the same as the DBA user, then the server pool can be created at the time DBCA is run by selecting the Policy-managed option within DBCA. The server pool can also be created after installation by using Server Control (SRVCTL).

The minimum size of this server pool is the number of database instances. If the maximum size of the server pool is greater than the minimum size, then new instances can be added to the database to handle peak workloads or to accommodate growth.

For administrator-managed databases, the databases are automatically configured to run in the Generic server pool.

2. For policy-managed databases, the DBA creates an Oracle RAC database in the allocated server pool by selecting the Policy-managed option within DBCA.

3. The DBA creates database services that are managed by Oracle Clusterware. The application users connect to the database using these services.
4. The DBA enables the database for Oracle Database QoS Management using Enterprise Manager Cloud Control.

The initial configuration tasks for the Oracle Database QoS Management administrator are covered in more detail in the following sections.

For Oracle Solaris platforms, there is an additional consideration: [About Multi-CPU Binding on Solaris and Quality of Service Management](#)

- [Installing and Configuring Oracle Grid Infrastructure for a Cluster](#)
- [Creating and Configuring Server Pools](#)
By default, a server pool called the Free pool is created during Oracle Grid Infrastructure installation. To create server pools for your Oracle RAC database, you can use SRVCTL or Oracle Enterprise Manager.
- [Creating and Configuring an Oracle RAC Database](#)
The steps for creating and configuring an Oracle RAC database are not covered in this book.
- [Creating Oracle Database QoS Management Administrator Accounts](#)
Before logging in to the Oracle Database QoS Management Dashboard (the Dashboard), you must create an Oracle Database QoS Management administrative user. The operating system user associated with this account must be a cluster administrator user to initially set this up.
- [Enabling Oracle Database QoS Management](#)
If you have multiple databases running on the same cluster, you can specify which databases are managed by Oracle QoS Management.
- [About Multi-CPU Binding on Solaris and Quality of Service Management](#)
Using Multi-CPU Binding (MCB) and Oracle Database Quality of Service (QoS) Management together requires close communication between the system administrator and the database administrator (DBA).

8.1.1 Installing and Configuring Oracle Grid Infrastructure for a Cluster

The installation and configuration of Oracle Grid Infrastructure for a cluster is not covered in this book.

Related Topics

- *Oracle Grid Infrastructure Installation and Upgrade Guide*

8.1.2 Creating and Configuring Server Pools

By default, a server pool called the Free pool is created during Oracle Grid Infrastructure installation. To create server pools for your Oracle RAC database, you can use SRVCTL or Oracle Enterprise Manager.

When you use DBCA to create an Oracle RAC database, Oracle recommends that you select policy-managed for the database, and choose the server pools which the database instances should run in. If you choose the create an administrator-managed Oracle RAC database, then the database runs exclusively in the Generic server pool, which is created during the installation of Oracle Grid Infrastructure.

If you use a cluster administrator that is separate from the database administrator, then only the cluster administrator user can create server pools. The cluster administrator then grants privileges on the server pools to the operating system user that owns the Oracle RAC installation.

 **Note:**

When creating a server pool for use with Oracle Database QoS Management, do not configure the `SERVER_NAMES` attribute (the `-servers` option of `srvctl add svrpool` or `srvctl modify svrpool` commands) for the server pool. Full resource management is not supported in such a configuration because Oracle Database QoS Management cannot change server pool sizes. This is the same limitation that exists for resource management of administrator-managed databases.

 **See Also:**

- *Oracle Clusterware Administration and Deployment Guide* for information about server pools
- *Oracle Real Application Clusters Administration and Deployment Guide* for information about using SRVCTL to create a server pool
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for information about using DBCA to create an Oracle RAC database
- *Oracle Clusterware Administration and Deployment Guide* for more information about the cluster administrator user.

8.1.3 Creating and Configuring an Oracle RAC Database

The steps for creating and configuring an Oracle RAC database are not covered in this book.

When creating a database, Oracle recommends that you choose to create a policy-managed Oracle RAC database and specify the server pools in which it should run. If you create an administrator-managed Oracle RAC database, then it runs exclusively in the Generic server pool.

After you have created the databases, perform the following steps to configure the databases for use with Oracle Database QoS Management:

- [Modifying Database Initialization Parameters](#)
The `CPU_COUNT` parameter for each database instance that runs in a server pool must be set to the same value if the database is managed by Oracle Database QoS Management.
- [Creating Database Services](#)
Applications and users connect to the database using services.

Related Topics

- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*

8.1.3.1 Modifying Database Initialization Parameters

The `CPU_COUNT` parameter for each database instance that runs in a server pool must be set to the same value if the database is managed by Oracle Database QoS Management.

On each server, the sum of the values for `CPU_COUNT` for all database instances running on that server must be less than or equal to the physical CPU count. For example, if you have a server with eight CPUs, and there are two database instances running on this server, then, for the databases to be managed by Oracle Database QoS Management, the `CPU_COUNT` parameter for each database instance must be set so that the values of the `CPU_COUNT` parameters for all instances on the server add up to eight or less. For example, you could have `CPU_COUNT=3` on one instance and `CPU_COUNT=4` on the other instance, or `CPU_COUNT=6` on one instance and `CPU_COUNT=2` on the other instance.



Note:

By default, the CPU count of each database that is started on a server is set to the number of physical CPUs installed for that server.

If you are running more than one database in a server pool, then using the default settings for `CPU_COUNT` will cause Oracle Database QoS Management to report a violation. To avoid this error, manually configure the `CPU_COUNT` value in the SPFILE using either Oracle Enterprise Manager or SQL*Plus.

- Use SQL*Plus to modify the `CPU_COUNT` database initialization parameter for all instances of your Oracle RAC database:

```
ALTER SYSTEM SET cpu_count=n SCOPE=BOTH SID='*';
```

In the previous command, *n* is the number of CPUs that should be used by the database instances.

`CPU_COUNT` is a dynamic parameter that is not set by default. It should be set to the maximum number of CPUs that the database instance should utilize at any time. The sum of the values of `CPU_COUNT` for all instances on a server should not exceed the number of physical CPUs for that server. As a best practice, Oracle recommends using a minimum value of 2 for `CPU_COUNT`.

8.1.3.2 Creating Database Services

Applications and users connect to the database using services.

Details about creating database services are not included in this guide.

Related Topics

- *Oracle Real Application Clusters Administration and Deployment Guide*

8.1.4 Creating Oracle Database QoS Management Administrator Accounts

Before logging in to the Oracle Database QoS Management Dashboard (the Dashboard), you must create an Oracle Database QoS Management administrative user. The operating system user associated with this account must be a cluster administrator user to initially set this up.

The administrative user for the Oracle Database QoS Management server is referred to as the QoS Admin user. This user has access to all the features of the Oracle Database QoS Management server, including checking and changing the account password for the QoS Admin user. You can have multiple QoS Admin users.

1. As the cluster administrator user, log in to the node that is hosting the Oracle Database QoS Management server. This can be determined by using the following command from the Oracle Grid Infrastructure home:

```
srvctl status qosmsserver
```

2. Stop the Oracle Database QoS Management server resource.

```
srvctl stop qosmsserver
```

3. Log on as a CRS Administrator user and enter the following command:

```
qosctl qosadmin -setpasswd qosadmin
```

After you enter this command, you are prompted to enter the password of the default QoS Admin user one or more times.

If you want to use a different user name, then you would enter the following command:

```
qosctl qosadmin -adduser username
```

In this example:

- *qosadmin* is the name of the default QoS Admin user.
 - *username* is the name of the QoS Admin user you are creating. You are prompted to enter a password for this user
4. Restart the Oracle Database QoS Management server resource.

```
srvctl start qosmsserver
```

Related Topics

- [QOSCTL Utility Reference](#)
QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.
- [Creating Administrative Users for Oracle Database QoS Management](#)

8.1.5 Enabling Oracle Database QoS Management

If you have multiple databases running on the same cluster, you can specify which databases are managed by Oracle QoS Management.

You enable Oracle Database QoS Management in a hierarchical manner:

- Measuring, monitoring, or managing the cluster
- Measuring, monitoring, or managing individual databases that run on the cluster

To manage a database, all the databases that use the same user-defined server pool must be enabled for Oracle Database QoS Management if:

- One or more Performance Classes in that user-defined server pool are not marked "Measure-Only" in the active policy
- There are Performance Classes that include a service hosted by that database

If you do not enable all the databases in the same user-defined server pool for Oracle Database QoS Management and any of the above conditions exist, then a violation is signaled when you try to access the Dashboard for the database. If all of the Performance Classes in the user-defined server pool are in measure-only or monitor mode and none of the Performance Classes specify a hosted service, then there is no violation reported when accessing the Dashboard for the database.

 **Note:**

If you enable Oracle QoS Management to monitor or manage a container database (CDB), then all contained pluggable databases (PDBs) are monitored or managed as well. You cannot configure Oracle QoS Management to monitor or manage individual PDBs.

To enable Oracle QoS Management for your system, perform the following steps:

1. [Enable Oracle QoS Management at the Database Level](#)
2. [Create an Initial Policy Set](#)
3. [Enable Oracle QoS Management at the Cluster Level](#)
 - [Enable Oracle QoS Management at the Database Level](#)
 - [Create an Initial Policy Set](#)
 - [Enable Oracle QoS Management at the Cluster Level](#)

8.1.5.1 Enable Oracle QoS Management at the Database Level

1. Log in to Oracle Enterprise Manager Cloud Control as the database administrator.
2. From the Database targets page, select the database you want to modify.
3. Select **Availability**, then **Enable / Disable Quality of Service Management**.
4. Enter the Cluster and Database credentials, then click **Login**.

 **Note:**

To complete this step, you must specify the login information for both a SYSDBA and a cluster administrator account.

The Enable/Disable QoS Management screen is displayed.

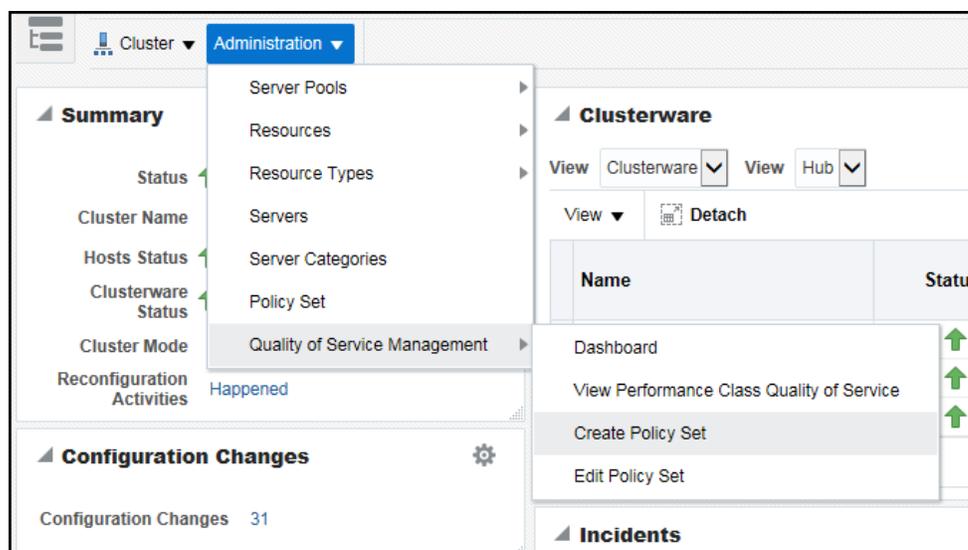
5. You are prompted to enter a password for the APPQOSSYS user. Choose a password and enter it in the Password and Confirm Password fields, then click OK.

When you provide a password, the following actions take place:

- The APPQOSSYS account, which enables the Oracle Database QoS Management server to connect to the database, is unlocked and the new password is set.
 - The credentials are written to an Oracle Wallet stored in the Oracle Cluster Registry to enable Oracle Database QoS Management to log in to the database.
6. APPQOS_PLAN is set as the active Oracle Database Resource Manager plan for all actively managed databases, so that Oracle Database QoS Management can adjust CPU access for Performance Classes. The APPQOS_PLAN is not required for databases where all their Performance Classes are checked Measure-Only.

8.1.5.2 Create an Initial Policy Set

1. On the Oracle Enterprise Manager Cloud Control All Targets page, select the cluster on which your cluster database that has Oracle Database QoS Management enabled runs.
2. Select **Administration**, then **Quality of Service Management**, then **Create Policy Set**.



3. Log in to the Oracle Database QoS Management Server using the QoS Management administrator password (default user name is qosadmin).
4. On the first page of the Create Policy Set wizard, check the Manage box next to the server pools that represent your database. For example, `online` and `backoffice`. Click **Next**.

Details	Server Pool	Services	Violation (s)	Server Pool Size			Importance	Manage
				Min	Max	Current		
Show	srvprod	prod:salesrpt,webdb:websvc,prod:manager,prod:hcmrpt,webdb:webprt,prod:employee,prod:sales	None	0	2	2	0	<input checked="" type="checkbox"/>
Show	Generic	dev:salesdev,test:salestest	None	0	-1	1	0	<input checked="" type="checkbox"/>
Show	Free	None	None	0	-1	0	0	<input checked="" type="checkbox"/>

- To get started with Oracle Database QoS Management, accept the defaults for your initial configuration and click **Next** on each page of the wizard to use the Default Policy Settings. On the fifth step, click **Set Policy** to set the DefaultPolicy as the Chosen Active Policy, then click **Next**.

On the last step of the Create Policy Set Wizard, click **Submit Policy Set**.

8.1.5.3 Enable Oracle QoS Management at the Cluster Level

- Using Oracle Enterprise Manager Cloud Control, on the All Targets page, select the cluster on which your cluster database that has Oracle Database QoS Management enabled runs.
- Select **Administration**, then **Quality of Service Management**, then **Dashboard**.
- Log in as the Oracle Database QoS Management user (for example, qosadmin).
- On the Dashboard page, the General section shows the current status of Oracle Database QoS Management. On a new system, the status is Disabled. Click the link **Disabled** next to the status to enable Oracle Database QoS Management for this cluster.

8.1.6 About Multi-CPU Binding on Solaris and Quality of Service Management

Using Multi-CPU Binding (MCB) and Oracle Database Quality of Service (QoS) Management together requires close communication between the system administrator and the database administrator (DBA).

Multi-CPU binding (MCB) is an Oracle Solaris projects resource management functionality that is used to bind a project to a specific set of CPUs, but not bind the CPUs exclusively. MCB allows other processes also to use these CPUs and allows overlapping of partitions. MCB is supported on Oracle Solaris 11.3. Control groups (CGroups) on Linux systems is another system administrator methods of managing server resources by allocating CPU and server resources to specific applications.

MCB has no impact on the use of Oracle Database Quality of Service (QoS) Management when used in measure and monitor mode. When you use Oracle Database Quality of Service (QoS) Management in management mode for a group of servers, there are four resource controls that Oracle QoS Management currently supports:

1. **Consumer Group Mappings:** CPU shares between competing workloads within a Non-CDB or PDB.
2. **Container Database (CDB) Resource Plans:** CPU Shares between competing PDBs within a CDB
3. **Instance Caging:** CPUs/Threads between co-hosted database instances
4. **Server Pool Cardinality:** number of servers in a server pool offering the database

MCB becomes a problem with regards to instance caging because it is possible for Oracle Database Quality of Service (QoS) Management to recommend a change in `CPU_COUNT` that would not be honored by the operating system. If the recommended action is implemented in this situation, there would probably still be some improvement to the target workload because the donor database would lose a CPU. This would cause Resource Manager to not schedule as many parallel sessions which would help out when hard partitioning is not used. However, the projected performance improvement would be overstated.

9

Administering the Oracle Database QoS Management System

This chapter describes the basic administrative tasks you perform when using Oracle Database QoS Management to manage performance of your Oracle RAC cluster.

- [Determining If Oracle Database QoS Management is Enabled](#)
- [Monitoring Performance with Oracle Database QoS Management](#)
- [Using the Oracle Database QoS Management Dashboard](#)
The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.
- [Administering the Policy Set](#)
Whether you are configuring the Oracle Database QoS Management system for the first time, or want to create a new Policy Set, you use the Create Policy Set wizard to create your Policy Set. You use the Edit Policy Set wizard to modify your existing Policy Set.
- [Managing Performance Classes](#)
- [Managing Performance Policies](#)
- [Reviewing Performance Metrics](#)
You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.
- [Creating Administrative Users for Oracle Database QoS Management](#)
- [Editing the Resource Plan for Oracle Database QoS Management](#)
You can perform limited editing of the Resource Manager plans used by Oracle Database Quality of Service Management.

9.1 Determining If Oracle Database QoS Management is Enabled

You enable Oracle Database QoS Management in a hierarchical-method. Levels 2 and 3 depend on the previous levels being configured.

1. On the cluster: Required for all operations
2. On the server pool: Required for any operation upon the node in that server pool
3. On the database: Required for any operation

If you have multiple databases within the same Oracle RAC cluster, they might not all be managed by Oracle Database QoS Management. If you enable Oracle Database QoS Management for a multitenant database, then Oracle Database QoS Management is enabled for all pluggable databases (PDBs) of that container database (CDB).

- [Checking the Enabled Status for a Database](#)
- [Checking the Enabled Status for the Cluster](#)

9.1.1 Checking the Enabled Status for a Database

To determine if your database is managed by Oracle Database QoS Management, perform the following steps:

1. Log in to Cloud Control and select the database target to check.
2. From the target's menu, select **Cluster Database**, then **Target Information**.
3. In the Target Information window, at the bottom, the value for QoS Status should be Enabled if Oracle Database QoS Management is enabled for this database.

9.1.2 Checking the Enabled Status for the Cluster

Using Oracle Enterprise Manager Cloud Control, you can determine whether Oracle Database QoS Management is enabled for your cluster.

To determine if a cluster is managed by Oracle Database QoS Management, perform the following steps:

1. Log in to Cloud Control and select the cluster target to check.
2. From the target's menu, select **Administration**, then **Quality of Service Management**, and then **Dashboard**.
3. If prompted, log in as the QoSAdmin user.
4. On the Dashboard, in the General section, check the value for QoS Status. If the status value is Enabled, then the cluster is being managed by Oracle Database QoS Management.



9.2 Monitoring Performance with Oracle Database QoS Management

After you have enabled Oracle Database QoS Management and created a default policy set, as described in "[Enabling Oracle Database QoS Management](#)", you can start to use Oracle Database QoS Management to monitor the performance of your system.

In Measure-only and Monitor modes, all the Performance Classes in user-defined Performance Policies have the measure-only box checked. You can set Performance Objectives, and Oracle Database QoS Management displays the Performance Satisfaction Metric (PSM) on the dashboard. If the response time of the system

exceeds the Performance Objective specified, the PSM bar changes to red and an optional alert generated, as shown in [Figure 9-1](#). Oracle Database QoS Management does not make recommendations if the measure-only check box is selected.

Figure 9-1 Performance Satisfaction Metrics for Measure-Only Performance Classes

Performance Overview						
This table provides an overview of Performance Class Metrics. Based on the collected metrics, QoS Management identifies a Target Performance Class and provides recommendations to help meet its Performance Objective.						
Performance Classes	Server Pools	Rank	Objective Type	Measure Only	Resource Use vs Wait Time (Last 5 sec)	Performance Satisfaction Metric (Last 5 min)
Default_pc	online,backoffice	Medium	Average Response Time	✓		
erp_pc	backoffice	Medium	Average Response Time	✓		
etl_pc	backoffice	Medium	Average Response Time	✓		
hr_pc	backoffice	Medium	Average Response Time	✓		
sales_pc	online	Medium	Average Response Time	✓		
salescart_pc	online	Medium	Average Response Time	✓		
shipping_pc	online	Medium	Average Response Time	✓		

TIP * next to Performance Class indicates that QoS Management is making recommendations for that Performance Class at this time

Running Oracle Database QoS Management in Measure-only or Monitor mode allows you to understand how various workloads perform when sharing resources. Only measure-only mode assists in determining the baseline Performance Objectives. In monitor mode, you can actually set the performance objectives, and monitor the workload performance against that performance objective. You can also use these modes to identify performance bottlenecks in your system.

Starting with the Oracle Database 12c release 2 (12.2.0.1) release, you can use Oracle Database QoS Management with Oracle RAC on systems in full management mode in both policy- and administrator-managed deployments. Oracle Database QoS Management also supports the full management of multitenant databases in both policy- and administrator-managed deployments. Earlier releases only support measure-only and monitor modes on Oracle RAC multitenant and administrator-managed deployments.

Related Topics

- [Reviewing Performance Metrics](#)
You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.
- [Managing Performance Classes](#)
- [Creating a Performance Policy and Specifying Performance Objectives](#)
Use Oracle Enterprise Manager Cloud Control to create a Performance Policy.

9.3 Using the Oracle Database QoS Management Dashboard

The Oracle Database QoS Management Dashboard (the Dashboard) provides an easy to use interface for managing the Oracle Database QoS Management system.

- [Accessing the Oracle Database QoS Management Dashboard](#)
The dashboard provides you with an overall view of the quality of service being delivered by the workloads in your system.

- [Enabling Oracle Database QoS Management for a Cluster](#)
- [Disabling Oracle Database QoS Management for a Cluster](#)
- [Interpreting the Performance Overview Graphs](#)
- [Viewing Recommendations](#)
- [Viewing Recommendation Details](#)
- [Implementing Recommendations](#)
By default, Oracle Database QoS Management does not automatically implement recommendations.

9.3.1 Accessing the Oracle Database QoS Management Dashboard

The dashboard provides you with an overall view of the quality of service being delivered by the workloads in your system.

The Dashboard has four main sections:

- General
- Performance Overview
- Recommendations
- Resource Wait Time Breakdown

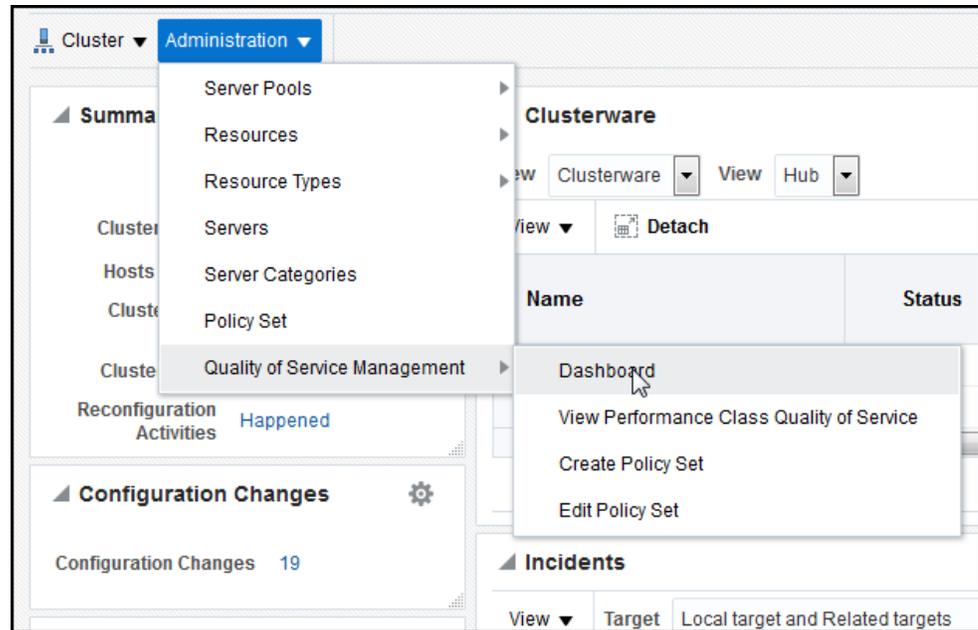
The General section of the Dashboard gives you a quick overview of the system. This section lists the QoS Status (Enabled or Disabled), the Current Active Policy, a link for checking the Performance Class details, and a notification for available recommendations. There is also a button that enables you to quickly change the current active policy.

In the Performance Overview section, there is a table that lists the Performance Classes, the server pools where work is occurring, their rank, the Performance Objective being measured, and whether the Performance Objectives are being monitored only, or are being monitored and managed. For each Performance Class there are bar graphs that provide an overview of the Performance Class metrics.

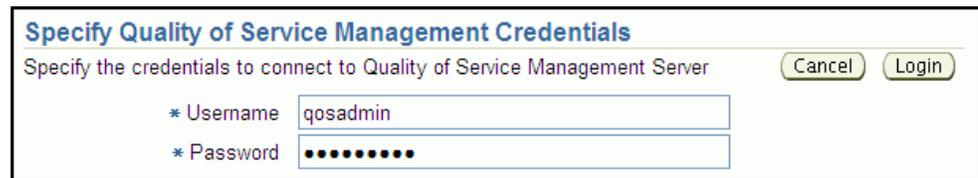
In the Recommendation section you can view the recommendations that are available when a Performance Class is not meeting its Performance Objectives. You can also view any violations that prevent the recommendations from being made.

At the bottom of the Dashboard is the Resource Wait Times Breakdown section. This section contains a table that provides a breakdown of resource wait times by Performance Class. For each Performance Class, the blocking resource is the one that has the most wait time. This data is used by QoS Management to produce Recommendations. The data can also be used to make manual adjustments to the system. If you expand each Performance Class listed in the table, then you can see the server pools for that Performance class, and the resource wait times for each server pool.

1. Log in to Cloud Control and select the cluster target to check.
2. From the target's menu, select **Administration**, then **Quality of Service Management**, and then **Dashboard**.

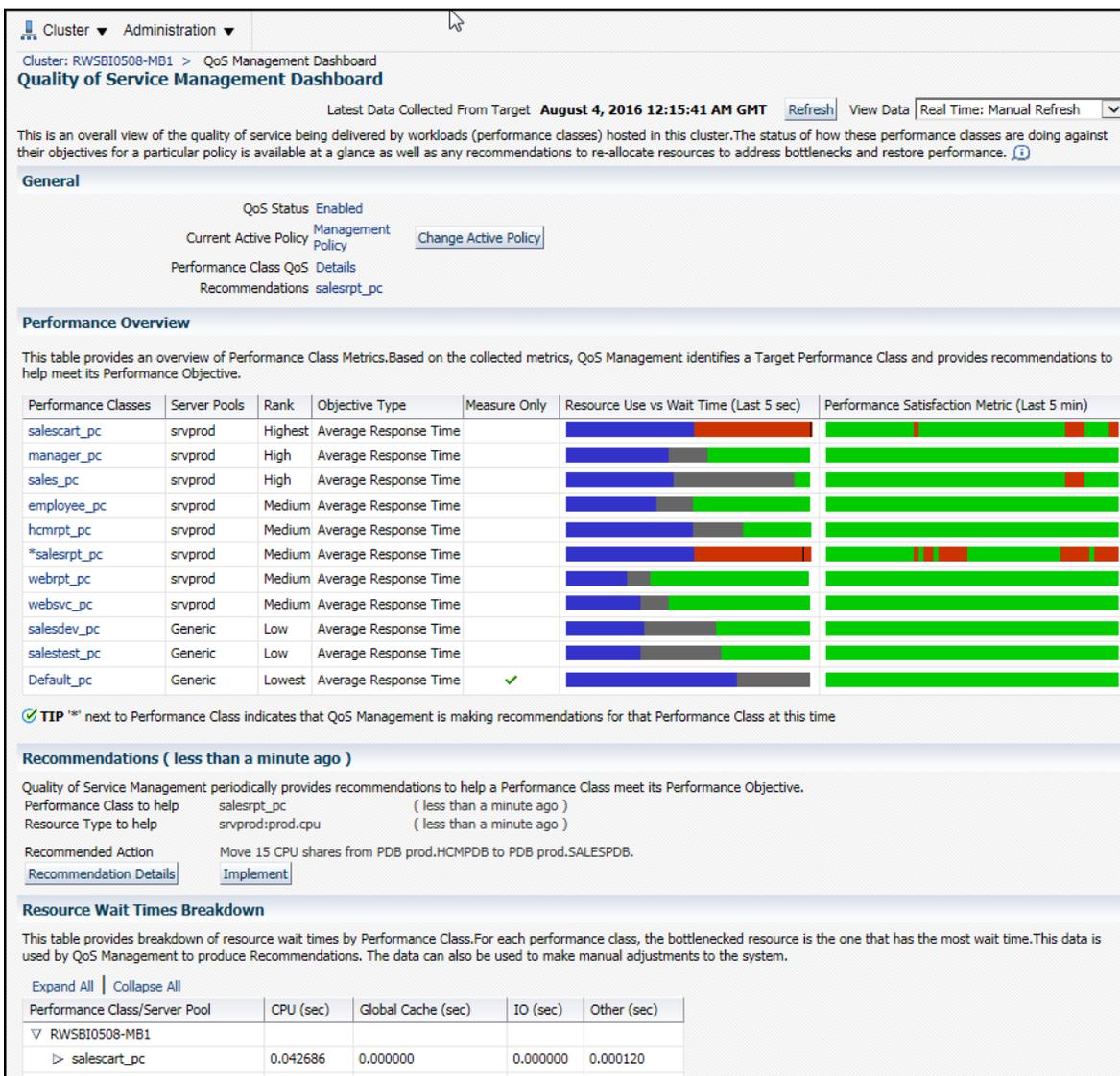


3. A login screen appears, prompting you for the Oracle Database QoS Management administrator (QoSAdmin) credentials. After typing in the user name and password, click **Login**.



4. The Quality of Service Management Dashboard page, shown in [Figure 9-2](#), is displayed after the correct credentials are entered.

Figure 9-2 Oracle Database Quality of Service Management Dashboard



Related Topics

- [Interpreting the Performance Overview Graphs](#)

9.3.2 Enabling Oracle Database QoS Management for a Cluster

1. On the Dashboard, in the General section, next to QoS Status, click **Disabled**.
2. On the Enable / Disable Quality of Service Management page, click **Enable QoS Management**.

9.3.3 Disabling Oracle Database QoS Management for a Cluster

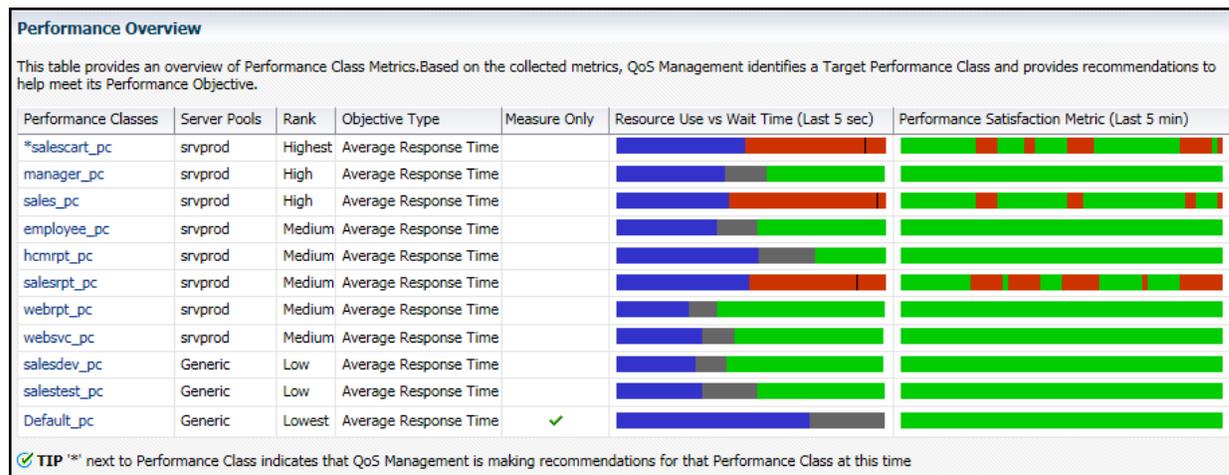
1. On the Dashboard, in the General section, next to QoS Status, click **Enabled**.

2. On the Enable / Disable Quality of Service Management page, click **Disable QoS Management**.

9.3.4 Interpreting the Performance Overview Graphs

On the Dashboard, in the Performance Overview section, there is a list of the current Performance Classes, some basic information about each Performance Class, and two bar graphs that show the most recent trend for the performance metrics for that class, as shown in [Figure 9-3](#).

Figure 9-3 Example of the Performance Overview Graphs



[Figure 9-3](#) shows the Performance Overview section of the Dashboard, which consists of a table with embedded bar graphs. The rows in the table list each performance class, the server pools associated with that performance class, its rank and objective types. Also included in the row for each performance class are two bar graphs, one showing the resource use compared to the wait time, and the other graph showing the performance satisfaction metric over the last 5 minutes. The Resource Use vs. Wait Time bar graph has three sections of varying size and color that illustrate the resource usage (blue), wait time (gray or red), and headroom (green) portions of the Performance Objective. The Performance Satisfaction Metric bar graph is displayed as a single bar, growing from left to right, with the green and red segments representing five second time slices in which the performance class was either exceeding (green) or violating (red) its performance objectives. If you place your cursor over a section of the Resource Use vs. Wait Time bar graph, then a description of that measurement appears by your cursor.

Resource Use vs. Wait Time

The Resource Use vs. Wait Time graph is refreshed only when you refresh the page contents. In this graph:

- The blue section represents the portion of the average time spent for all database requests by that Performance Class that are using resources in the last five seconds

- The gray section represents the portion of the average time spent for all database requests by that Performance Class that are waiting on resources in the last five seconds
- The green section represents the headroom for that Performance Class (proportion of the average time for all database requests below the specified Performance Objective) in the last five seconds
- If a Performance Class is not meeting its Performance Objectives, then the gray and green sections disappear and the resource wait time is shown in red with a line to indicate where the Performance Objective is relative to the actual response time.
- If you place your mouse cursor over any section of this bar graph, then the actual values of Use, Wait and Headroom are displayed

The point between the gray and the green sections of the bar graph is the Performance Objective value. If you set this value below the resource use time, then you will never meet that objective. When configuring the Performance Objectives for a Performance Class, you must set the Performance Objective high enough to produce sufficient headroom (shown in green) to be able to share resources between Performance Classes to meet service levels as demand changes.

If a red section appears in the bar graph for a Performance Class, then you know that the Performance Class is not meeting its Performance Objectives. Oracle Database QoS Management issues a recommendation and an action to implement, if possible, to correct the problem. The recommendations generated by Oracle Database QoS Management occur once each minute, so they correspond to an earlier time than the current Performance Overview graphs.

Performance Satisfaction Metric

The Performance Satisfaction Metric changes to show red and green segments for specific five second samples. Using this you can spot trends in the performance of your system.

For the Performance Satisfaction Metric bar graph:

- The red section represents the periods of time the Performance Class was not meeting its Performance Objectives during the sampling period
- The green section represents the periods of time the Performance Class was meeting its Performance Objectives during the sampling period

9.3.5 Viewing Recommendations

When viewing recommendations, there are three possible results.

1. If the Performance Classes are meeting their Performance Objective, then a recommendation is displayed which states "No action required: all Performance Objectives are being met."
2. If Oracle Database QoS Management determines that a Performance Class is not meeting its Performance Objective and has a recommended action for improving performance, then the Dashboard places an asterisk (*) in front of the Performance Class name in the Performance Overview chart and displays a Recommended Action.
3. If more than one Performance Class is not meeting its Performance Objective as shown in [Figure 9-4](#), then only the target Performance Class displays an asterisk

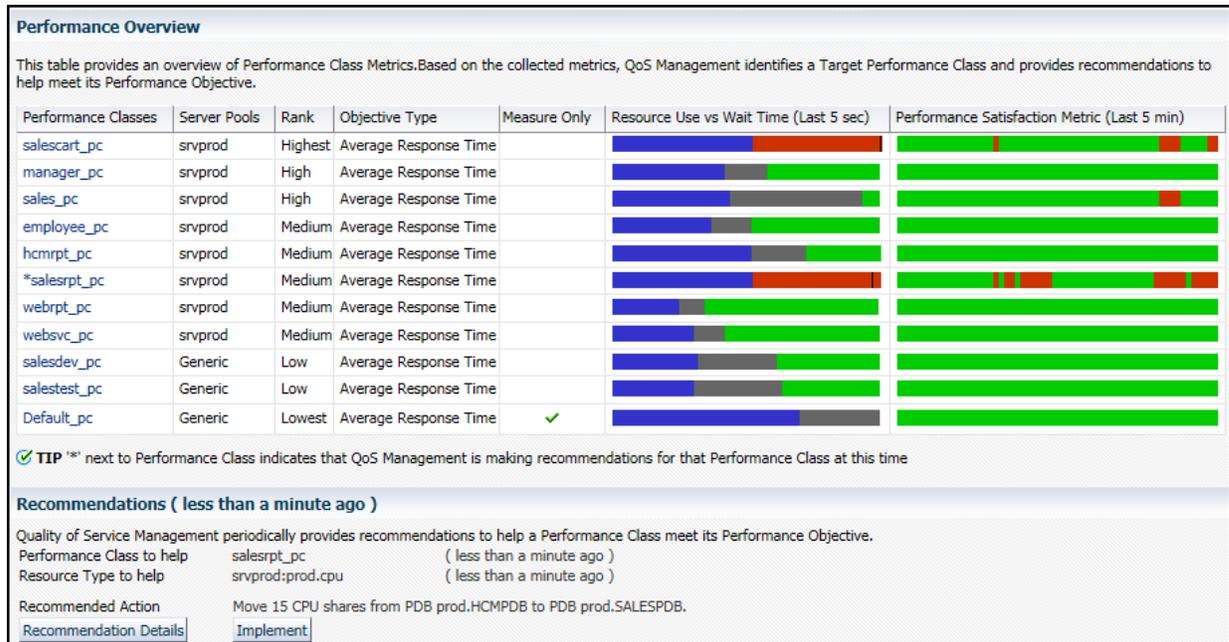
and a recommendation. If the recommendation has an associated action, then an **Implement** button appears, which you can click to have the action implemented.

If there are no recommended actions, then you can click the Details button to view the results of the latest analysis. Oracle Database QoS Management shows what possible actions could be taken, and why these actions were not chosen.

Example 9-1 Example Output For Performance Overview Page and Recommended Actions

The screenshot shows two performance classes are not meeting their performance objectives, salescart_pc and salesrpt_pc. As a result, Oracle Database QoS Management has generated a recommended action to move 15 CPU shares from the pluggable database prod.HCMPDB to the pluggable database prod.SALESPDB. Implementing this action will provide more CPU to the workload associated with the salescart_pc performance class.

Figure 9-4 Performance Classes Not Meeting Their Performance Objectives



9.3.6 Viewing Recommendation Details

If Oracle Database QoS Management has generated a recommendation, then you can click the **Recommendation Details** button to view more information about why the recommendation was made, and the expected performance improvements to be gained if you implement the recommendations.

Recommended Actions

Action: Rank 1: Demote sales_pc from Consumer Group 2 to Consumer Group 3.

Action: Demote sales_pc from Consumer Group 2 to Consumer Group 3.

Estimated Time: approximately 2 minutes

Rationale: All potential single mapping changes have been analyzed. Changes evaluated and rejected are listed below.

Evaluation: The beneficiary's PSM value is expected to change by 23.55 percentage points. The sum of all PSM values is expected to change by -2.347 percentage points. This action is a candidate for recommendation.

Projected Results

Performance Class	Performance Satisfaction Metric (Last 5 min)		Average Response Time		
	Projected (%)	Projected Change (%)	Objective Value (sec)	Current Value (sec)	Projected Value (sec)
Default_pc	100	0.0	0.00000	0.00113	0.00113
salestest_pc	43	0.0	0.10000	0.05738	0.05738
salesdev_pc	58	0.0	0.10000	0.04227	0.04227
websvc_pc	55	0.0	0.10000	0.04526	0.04526
webprt_pc	62	0.0	0.12000	0.04522	0.04522
hcmrpt_pc	25	0.0	0.10000	0.07478	0.07478
salesrpt_pc	19	23.3	0.10000	0.10455	0.08100
employee_pc	48	0.0	0.05000	0.02617	0.02617
sales_pc	-47	-49.2	0.06000	0.05893	0.11420
manager_pc	44	0.0	0.05000	0.02803	0.02803
salescart_pc	22	23.5	0.09000	0.09122	0.07001

Implement

You can select different recommendations using the Action drop-down list. Oracle Database QoS Management tries to provide the best recommendation to the QoS administrator, but you could decide that a different action would produce quicker results. If you select a different action, then the information in the Recommendation Actions and the Situation Analysis sections are updated to reflect the impact of the alternate recommendation. You cannot implement an alternate recommendation if that recommendation has been rejected by the Oracle Database QoS Management System for not providing enough benefit to the system as a whole.

Recommended Actions

Action: Rank 3: Move 1 CPU in Server Pool srvprod from Slice webdb to Slice prod.

Action: Move 1 CPU in Server Pool srvprod from Slice webdb to Slice prod.

Estimated Time: approximately 3 minutes

Rationale: All potential single CPU changes have been analyzed.

Evaluation: The beneficiary's PSM value is expected to change by 17.579 percentage points. The sum of all PSM values is expected to change by -88.082 percentage points. This action is a candidate for recommendation.

The Recommended Actions page also has a section called Situation Analysis. If you display the output in this section, then you can see a description the projected impact implementing the Recommendation will have on the Performance Classes and the server pool, as shown in the following screenshot.

▽ Situation Analysis	
Donor Performance Classes	Donor Server Pools
Quality of Service Management could help salescart_pc at the expense of sales_pc: sales_pc is another PC using resource cpu in Server Pool online. sales_pc's Performance Objective is of lesser rank than salescart_pc's Performance Objective.	Quality of Service Management could move servers from Server Pool backoffice to Server Pool online: The current size of Server Pool backoffice is larger than its configured minimum size.
Quality of Service Management could help salescart_pc at the expense of Default_pc: Default_pc is another PC using resource cpu in Server Pool online. Default_pc is not currently violating its Performance Objective. Default_pc's Performance Objective is of lesser rank than salescart_pc's Performance Objective.	

9.3.7 Implementing Recommendations

By default, Oracle Database QoS Management does not automatically implement recommendations.

1. If there is a Recommended Action, and you decide to implement the action, then simply click the **Implement** button on the Dashboard or the Details page.

After you have implemented a recommendation, the display on the Dashboard changes to show that there is an action in progress. No new recommendations are displayed until the system has settled after the resource change. You can determine the amount of time you have to wait before any new recommendations are made available by viewing the Details page before you implement a recommendation.

2. After implementing a Recommended Action, view the Performance Overview charts on the Dashboard to determine if the Performance Class is now meeting its Performance Objectives.

Related Topics

- [Automatically Implementing Recommendations for a Performance Policy](#)

9.4 Administering the Policy Set

Whether you are configuring the Oracle Database QoS Management system for the first time, or want to create a new Policy Set, you use the Create Policy Set wizard to create your Policy Set. You use the Edit Policy Set wizard to modify your existing Policy Set.

Note:

If you decide to create a new Policy Set, then all existing Performance Policies and user-added Performance Classes must be re-created. The changes you make to the stored Policy Set are not saved until you click **Submit Policy Set** in the last page of the Policy Set wizard.

- [Editing a Policy Set](#)
You can modify the policy sets for your system.

- [Adding Server Pools to a Policy Set](#)
Server pools are created by the cluster or database administrator, using either Oracle Enterprise Manager Cloud Control or Server Control (SRVCTL).
- [Modifying Server Pool Settings](#)
To modify the server pool settings, you configure a server pool directive override for a Policy Set.
- [Adding Database Services to a Policy Set](#)
By creating additional services, you can monitor your workload performance at a finer level, by limiting use of the new services to specific applications or users.
- [Updating a Policy Set to Include a New Database](#)
If a new database is added to your cluster, you can configure Oracle Database QoS Management to manage or monitor the workload on this database.

9.4.1 Editing a Policy Set

You can modify the policy sets for your system.

To modify an existing Policy Set, perform the following steps:

1. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then **Edit Policy Set**.

The Policy Set Editor wizard is started.

2. On the first page of the Policy Set Editor wizard you can view the current server pool settings.

Details	Server Pool	Services	Violation (s)	Server Pool Size			Importance	Manage
				Min	Max	Current		
▶ Show	Generic	dev:salesdev, test:salestest	None	0	-1	1	0	<input checked="" type="checkbox"/>
▶ Show	srvprod	prod:salesrpt, webdb:websvc, prod:manager, prod:hcmrpt, webdb:webbrpt, prod:employee, prod:sales	None	0	2	2	0	<input checked="" type="checkbox"/>
▶ Show	Free	None	None	0	-1	0	0	<input checked="" type="checkbox"/>

✔ TIP -1 for Max indicates whole cluster

This page lets you specify which server pools are managed by Oracle Database QoS Management. If the Manage box for a server pool is unchecked, then none of the servers, databases or workloads that use that server pool are displayed or managed by Oracle Database QoS Management. The Details column displays any configuration violation details and the corrective action to take to enable the server pool to be managed.

When finished, click **Next**.

The Policy Set Editor: Performance Classes page appears.

3. The second page enables you to create, edit, rename, or delete Performance Classes for the system. You can create a new Performance Class, or use the Edit Performance Class button to modify the classifiers for an existing Performance Class.

Edit Policy Set: Performance Classes

Performance Classes are a collection of work requests for which performance objectives wish to be set. [i](#)

[Add Performance Class](#) | [Edit Performance Class](#) | [Rename Performance Class](#) | [Delete Performance Class](#)

Expand All | Collapse All

Select	Cluster / Performance Class / Classifiers
<input checked="" type="radio"/>	▼ RWSBI0508-MB1
<input type="radio"/>	▶ salestest_pc
<input type="radio"/>	▶ employee_pc
<input type="radio"/>	▶ salesdev_pc
<input type="radio"/>	▶ sales_pc
<input type="radio"/>	▶ Default_pc
<input type="radio"/>	▶ salescart_pc
<input type="radio"/>	▶ hcmrpt_pc

You can use the **Expand All** link to show the classifiers for each Performance Class, or expand an individual Performance Class entry to show the classifiers for only that Performance Class.

Select the Performance Class you want to edit, and then click the appropriate action button (**Edit Performance Class**, **Rename Performance Class**, or **Delete Performance Class**). If you want to create a Performance Class for this Policy Set, then click **Add Performance Class**.

When you have finished, click **Next**.

4. After creating or modifying the Performance Classes, the next step is to set the order in which the classifiers are evaluated.

Edit Policy Set: Classifier Ordering

Work requests are evaluated against the Classifier's Boolean expression in the order specified below. [i](#)

Select	Performance Class	Classifier
<input checked="" type="radio"/>	salescart_pc	SERVICE_NAME INSET prod:sales,USER INSET SOES
<input type="radio"/>	employee_pc	SERVICE_NAME INSET prod:employee
<input type="radio"/>	hcmrpt_pc	SERVICE_NAME INSET prod:hcmrpt
<input type="radio"/>	manager_pc	SERVICE_NAME INSET prod:manager
<input type="radio"/>	sales_pc	SERVICE_NAME INSET prod:sales
<input type="radio"/>	salesdev_pc	SERVICE_NAME INSET dev:salesdev
<input type="radio"/>	salesrpt_pc	SERVICE_NAME INSET prod:salesrpt
<input type="radio"/>	salestest_pc	SERVICE_NAME INSET test:salestest
<input type="radio"/>	webrpt_pc	SERVICE_NAME INSET webdb:webrpt
<input type="radio"/>	websvc_pc	SERVICE_NAME INSET webdb:websvc

This step is very important, because the classifiers determine which Performance Class a work request is placed into. Select a Performance Class and use the arrow keys to the right of the classifiers to move the entries up and down in the list.

As a work request enters the system, the work request is evaluated against the classifiers. The first expression that evaluates to `TRUE` determines which Performance Class the work request belongs to. You should put the more fine-grained classifiers at the top, and the more generic classifiers at the bottom of the list.

- After you have set the order for your Performance Classes, the next step is to edit the Performance Policies and set the Performance Objectives.

Edit Policy Set: Performance Policies

A Performance Policy is a collection of performance objectives, one for each Performance Class, that are in force together. [?](#)

[Add Policy](#)

[Copy Policy](#) [Edit Policy](#) [Delete Policy](#)

Select	Name	Policy Description
<input checked="" type="radio"/>	DefaultPolicy	This is the default policy.
<input type="radio"/>	Measurement Policy	This is the measurement policy.
<input type="radio"/>	Monitor Policy	This is the monitor policy.
<input type="radio"/>	Management Policy	This is the management policy.

On this page you can:

- Create a new Performance Policy
- Edit, copy, or delete an existing Performance Policy
- Specify which resource allocation methods can be automatically implemented by QoS Management.

 **Note:**

The DefaultPolicy Performance Policy cannot be modified, but the policy can be copied and used as the basis for a new Performance Policy.

- After you have configured the Performance Policies, you must choose which one to put into effect immediately after the Policy Set is submitted.

Edit Policy Set: Set Policy

Select the policy which will initially be in effect when the policy set is submitted. [?](#)

Current Active Policy Management Policy
Chosen Active Policy Management Policy

[Set Policy](#)

Select	Name	Policy Description
<input type="radio"/>	DefaultPolicy	This is the default policy.
<input type="radio"/>	Measurement Policy	This is the measurement policy.
<input type="radio"/>	Monitor Policy	This is the monitor policy.
<input checked="" type="radio"/>	Management Policy	This is the management policy.

Select the preferred Performance Policy, then click **Set Policy**. When finished, click **Next**.

7. Before you submit a Policy Set to the Oracle Database QoS Management system, you are asked to review the Policy Set configuration.
 - If the changes are what you intended, then click **Submit Policy Set**.
 - If you must modify any of the displayed settings, then click the **Back** or **Cancel** button.

After clicking **Submit Policy Set**, you are returned to the Dashboard.

Related Topics

- [Creating a Performance Class](#)
- [Managing Performance Policies](#)

9.4.2 Adding Server Pools to a Policy Set

Server pools are created by the cluster or database administrator, using either Oracle Enterprise Manager Cloud Control or Server Control (SRVCTL).

- Refer to *Oracle Clusterware Administration and Deployment Guide* for instructions on how to create a server pool.
- After the cluster administrator has created a server pool, you can add the server pool to the Oracle Database QoS Management system.

Related Topics

- [Creating a Performance Policy and Specifying Performance Objectives](#)
Use Oracle Enterprise Manager Cloud Control to create a Performance Policy.

See Also:

- *Oracle Clusterware Administration and Deployment Guide* for more information about modifying the size of a server pool
- *Oracle Real Application Clusters Administration and Deployment Guide* for more information about configuring a recently allocated server to be a part of an existing Oracle RAC database.

9.4.3 Modifying Server Pool Settings

To modify the server pool settings, you configure a server pool directive override for a Policy Set.

When you configure server pool directive overrides, the new parameters you specify for the server pool are used instead of the parameters specified at the time of server pool creation.

 **Note:**

You can change the minimum and maximum settings for the listed server pools, but you cannot create additional server pools using this interface. Any changes made here, after they are submitted, alter the current server pool properties set in the Manage Server Pools section of Oracle Enterprise Manager Cloud Control when the associated Performance Policy is active.

- You can modify the server pool parameters manually using the Manage Server Pool pages of Oracle Enterprise Manager Cloud Control or SRVCTL.

You should not use this method of altering the server pool configuration when you have Oracle Database QoS Management enabled. If you use both server pool directive overrides and manually change the server pool configuration, then the server pool directive overrides specified for the current Policy Set will override the manual settings and can result in confusion.

Related Topics

- [Setting Server Pool Directive Overrides](#)
A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.

9.4.4 Adding Database Services to a Policy Set

By creating additional services, you can monitor your workload performance at a finer level, by limiting use of the new services to specific applications or users.

To add database services to a Policy Set, you must create or modify Performance Classes for each new database service.

1. The database administrator uses SRVCTL or Oracle Enterprise Manager Cloud Control to create database services that are managed by Oracle Clusterware. You can use the Availability menu on the database target page of Enterprise Manager Cloud Control to create services for an Oracle RAC database.
2. The QoS Administrator adds one or more Performance Classes to the Policy Set or modifies the existing Performance Classes to include the new database services

 **Note:**

Database services appear in Oracle Database QoS Management as `db_name:service_name`, so the service names must be unique within a database.

Related Topics

- *Oracle Database 2 Day + Real Application Clusters Guide*

9.4.5 Updating a Policy Set to Include a New Database

If a new database is added to your cluster, you can configure Oracle Database QoS Management to manage or monitor the workload on this database.

1. The database administrator first requests servers to host the cluster database from the cluster or system administrator.
2. The cluster or database administrator creates the new server pools with the specified minimum number of nodes.

The cluster or database administrator could also decide to grant access to existing server pools instead of creating new ones.
3. The database administrator creates a new database on the allocated server pools.

The database administrator must ensure that the new database is configured correctly for management by Oracle Database QoS Management.
4. The database administrator uses SRVCTL or Oracle Enterprise Manager Cloud Control to create database services that are managed by Oracle Clusterware for the new database.
5. The database administrator enables the database for Oracle Database QoS Management from the Database target page of Cloud Control.
6. Using the Edit Policy Set link in Cloud Control, the QoS Administrator adds the new server pools to the Policy Set, adds one or more Performance Classes to the Policy Set, or modifies the existing Performance Classes to include the new database services.
7. After you successfully submit the new Policy Set, the new database is monitored and managed by Oracle Database QoS Management.
8. You can update the Policy Set from the command-line using the option `qosctl qosadmin -updatepolicyset`. This is useful when adding a new database and/or a service in silent mode, or by running scripts.

Related Topics

- [Supported Database Configurations](#)
Depending on your database configuration, all or only some of the Oracle Database QoS Management features are available,
- [Editing a Policy Set](#)
You can modify the policy sets for your system.
- [QOSCTL Utility Reference](#)
QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.

9.5 Managing Performance Classes

Each policy set contains one or more performance classes. Each performance class defines a type of workload on your servers.

- [Creating a Performance Class](#)
- [Deleting a Performance Class](#)
You can delete a Performance Class that is no longer needed.

- [Renaming a Performance Class](#)
You can rename a Performance Class using the Edit Policy Set wizard.
- [Editing an Existing Performance Class](#)
When editing Performance Classes, you can create, edit, rename, or delete performance classes for the system. You can use the Edit Performance Class button to modify the classifiers for an existing performance class.
- [Specifying the Evaluation Order of the Classifiers](#)

9.5.1 Creating a Performance Class

To create or edit the Performance Classes and the classifiers they use, perform the following tasks:

1. Start the Policy Set Editor wizard. From the cluster target page in Oracle Enterprise Manager Cloud Control, select **Administration**, then **Quality of Service Management**, then **Edit Policy Set**.
2. Go to the second screen in the wizard.

On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed. If this is the first time configuring the system, then a Performance Class for each database service is shown along with a Default Performance Class.

Edit Policy Set: Performance Classes

Performance Classes are a collection of work requests for which performance objectives wish to be set.

[Add Performance Class](#) | [Edit Performance Class](#) | [Rename Performance Class](#) | [Delete Performance Class](#)

[Expand All](#) | [Collapse All](#)

Select	Cluster / Performance Class / Classifiers
<input checked="" type="radio"/>	▼ rwsbi0508-r
<input type="radio"/>	▶ sales_pc
<input type="radio"/>	▶ Default_pc
<input type="radio"/>	▶ etl_pc
<input type="radio"/>	▶ salesscart_pc
<input type="radio"/>	▶ shipping_pc
<input type="radio"/>	▶ hr_pc
<input type="radio"/>	▶ erp_pc

To specify a classifier for a work request, you must specify at least one service name. If you specify multiple services, then use a comma-delimited list. Optionally, you can also specify any of the following filters:

- A comma-delimited list of module names and whether the work request uses a module in this list

- A comma-delimited list of actions, and whether the work request performs an action in this list
 - A comma-delimited list of user names, and whether the work request uses a user name in this list
 - A comma-delimited list of programs, and whether the work request is running a program in this list
3. Click **Add Performance Class**, and the Performance Class creation page is displayed. In the Performance Class Name text field, enter a name for the Performance Class.

Add Performance Class

The Performance Class Name is used to identify a collection of work requests that flow through the system and against which performance objectives wish to be set.

Performance Class Name

Classifiers

Classifiers are a set of rules to classify an incoming request based upon a Boolean expression made up of database session parameters. Comma-delimited multiple values are allowed.

Select All | Select None

Select	Service	Module	Action	UserName	Program
<input type="checkbox"/>	<input type="text"/>	<input checked="" type="radio"/> In Set <input type="radio"/> Not In Set	<input checked="" type="radio"/> In Set <input type="radio"/> Not In Set	<input checked="" type="radio"/> In Set <input type="radio"/> Not In Set	<input checked="" type="radio"/> In Set <input type="radio"/> Not In Set

4. In the Classifiers section, enter information to define a rule for classifying work requests. First select a database service, then specify matching values (In Set) for the **module**, **action**, **UserName**, or **program name** that is associated with the work request using the specified database service. You can also specify exclusion values (Not In Set) for these attributes.

If you want to add multiple classifiers for the Performance Class, then click the **Add Classifier** button and enter in the appropriate information.

 **Note:**

When evaluating a classifier for a Performance Class, all of the specified values are compared to the work request attributes using an AND operation; if you specify multiple classifiers for the Performance Class, then the results of each classifier evaluation for that Performance Class are combined using an OR operation.

5. After you have defined all the classifiers for the Performance Class, click the **Next** button until you reach the end of the wizard. Review the information you specified, then click **Submit Policy Set**.

9.5.2 Deleting a Performance Class

You can delete a Performance Class that is no longer needed.

1. Start the Edit Policy wizard.
2. Go to the second screen in the wizard.
On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select a Performance Class and click **Delete Performance Class**.
4. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.5.3 Renaming a Performance Class

You can rename a Performance Class using the Edit Policy Set wizard.

1. Start the Edit Policy wizard.
2. Go to the second screen in the wizard.
On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select a Performance Class and click **Rename Performance Class**.
4. On the Rename Performance Class page, enter the new name of the Performance Class, then click **OK**.
5. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.5.4 Editing an Existing Performance Class

When editing Performance Classes, you can create, edit, rename, or delete performance classes for the system. You can use the Edit Performance Class button to modify the classifiers for an existing performance class.

1. Start the Policy Set Editor wizard.
 - a. From the cluster target page in Oracle Enterprise Manager Cloud Control, select **Administration**.
 - b. Select **Quality of Service Management**.
 - c. Select **Edit Policy Set**.
2. Go to the second screen in the wizard.
On the Edit Policy Set: Performance Classes page, the available Performance Classes are displayed.
3. Select the Performance Class you want to modify and click **Edit Performance Class**.
4. When you are finished making changes, click **OK**.
5. Advance to the last page of the Policy Set Editor and click **Submit Policy Set**.
 - [Adding Classifiers](#)
 - [Changing Classifiers](#)
 - [Deleting Classifiers](#)

9.5.4.1 Adding Classifiers

If you want to add a classifier to a Performance Class, then perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class to modify, then click **Edit Performance Class**.
4. In the Classifiers section on the Edit Performance Class page, click the **Add Classifier** button and enter in the appropriate information. When finished, click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.5.4.2 Changing Classifiers

If you want to modify one or more classifiers for a Performance Class, then perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class for which you want to change the classifiers, then click **Edit Performance Class**.
4. In the Classifiers section on the Edit Performance Class page, modify the classifier information, then click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.5.4.3 Deleting Classifiers

To delete one or more classifiers for a Performance Class, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Go to the second screen in the wizard, the Edit Policy Set: Performance Classes page.
3. Select the Performance Class for which you want to delete the classifiers, then click **Edit Performance Class**.
4. In the Classifiers section on the Edit Performance Class page, select the classifiers you want to delete, then click **Delete Classifiers**. When finished, click **OK**.
5. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.5.5 Specifying the Evaluation Order of the Classifiers

The classifiers generate Boolean expressions that are evaluated each time a work request enters the system. The first classifier that evaluates to TRUE determines the Performance Class for that work request. To ensure that the work requests are put in the correct Performance Classes, you must be careful in specifying the order in which the classifiers are evaluated.

To set the order of evaluation for the classifiers, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the third page in the wizard, which is titled Edit Policy Set: Classifier Ordering.
3. Use the arrow keys to the right of a classifier to move the classifier up or down in the list. The classifiers for the Performance Classes at the top of the list are evaluated first. If the work request does not match the classifiers for that Performance Class, then evaluation continues with the next Performance Class in the list, until there are no further evaluations to be made. If a work request matches the classifiers for a Performance Class, then the work request is associated with that Performance Class and evaluation stops.

For proper classification of work requests, you should put the Performance Classes with the strictest classifiers at the top of the list, and the Performance Classes with most lax classifiers near the bottom of the list. The `Default_pc` Performance Class, which has the most general classifiers, should always be at the bottom of the list.

4. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.6 Managing Performance Policies

- [Creating a Performance Policy and Specifying Performance Objectives](#)
Use Oracle Enterprise Manager Cloud Control to create a Performance Policy.
- [Editing an Existing Performance Policy](#)
On the Edit Policy page, you can change the rank of each Performance Class, or change the Measure Only setting for a Performance Class.
- [Copying a Performance Policy](#)
- [Setting the Current Performance Policy](#)
- [Deleting a Performance Policy](#)
- [Automatically Implementing Recommendations for a Performance Policy](#)
- [Setting Server Pool Directive Overrides](#)
A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.

9.6.1 Creating a Performance Policy and Specifying Performance Objectives

Use Oracle Enterprise Manager Cloud Control to create a Performance Policy.

To create and configure a Performance Policy, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.
3. Click the **Add Policy** button to create a Performance Policy. The Add Policy page appears.

Add Policy

Policy Name

Policy Description

Performance Definition for Policy:

Specify the business ranking and performance objectives for the Performance Classes below. By checking "Measure Only" a Performance Class will be monitored but not managed by QoS Management.

Performance Class	Rank	Objective Type	Objective Value (sec)	Current Value (sec)	Measure Only
sales_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
Default_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
etl_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
salescart_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
shipping_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
hr_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>
erp_pc	Medium ▼	Average Response Time	<input type="text" value="0"/>	0	<input checked="" type="checkbox"/>

Authorized Actions

Select types of resource allocation actions that may be automatically implemented by QoS Management.

Promote or Demote a Performance Class Consumer Group. Move a CPU between databases within a server pool. Move a server between server pools.

▼ **Server Pool Directive Override**

Server Pool Directive Overrides change the server pool availability properties when the associated policy is in effect. ⓘ

Server Pool	Server Pool Size						
	Min		Max		Current	Importance	
	Current Value	Override	Current Value	Override		Current Value	Override
backoffice	1	<input type="text" value="1"/>	2	<input type="text" value="2"/>	2	10	<input type="text" value="10"/>
Free	0	<input type="text" value="0"/>	-1	<input type="text" value="-1"/>	0	0	<input type="text" value="0"/>

For each Performance Policy you must specify a unique name. You can also provide a description of the policy and its intent. Then you must configure the Performance Classes for the policy.

To configure the Performance Classes, you must do the following:

- Set the rank for each Performance Class, from highest to lowest. A higher rank gives that Performance Class higher priority when there is contention for resources.

- Specify a value for the Performance Objective.

The Performance Objective value is the appropriate length of time in seconds in which the work request, or database request, should complete, for example, 0.008 seconds, or eight milliseconds.

 **Note:**

You should not use the service-level agreements (SLAs) or target response times as the Performance Objective values. Instead, choose a value that is reasonable, sustainable, and greater than your target response time. Using a higher value gives you time to implement the recommendations from Oracle Database QoS Management regarding the allocation of resources before the Performance Objective is exceeded.

You can also select whether the Performance Class is only measured or monitored, not managed. If you select Measure Only, then Oracle Database QoS Management measures or monitors the Performance Class, but does not provide any recommendations for improving performance.

4. Optional: Specify which actions can be performed automatically by QoS Management.
5. Optional: Configure server pool directive overrides.
6. Advance to the end of the Edit Policy Set wizard. Click **Submit Policy Set** to make the change permanent.

Related Topics

- [Automatically Implementing Recommendations for a Performance Policy](#)
- [Setting Server Pool Directive Overrides](#)
A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.
- [Monitoring Performance with Oracle Database QoS Management](#)

9.6.2 Editing an Existing Performance Policy

On the Edit Policy page, you can change the rank of each Performance Class, or change the Measure Only setting for a Performance Class.

You can also select the type of resource allocation actions that can be automatically implemented by QoS Management and set server pool directive overrides.

1. On the Edit Policy Set: Performance Policies page, select the Performance Policy you want to modify and click **Edit**.

Edit Policy

Policy Name

Policy Description

Performance Definition for Policy: DayTime

Specify the business ranking and performance objectives for the Performance Classes below. By checking "Measure Only" a Performance Class will be monitored but not managed by QoS Management.

Performance Class	Rank	Objective Type	Objective Value (sec)	Current Value (sec)	Measure Only
sales_pc	High ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.10000"/>	0.04213	<input type="checkbox"/>
Default_pc	Lowest ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.00000"/>	0.00468	<input checked="" type="checkbox"/>
etl_pc	Medium ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.09000"/>	0.03838	<input type="checkbox"/>
salescart_pc	Highest ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.10000"/>	0.06357	<input type="checkbox"/>
shipping_pc	High ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.09000"/>	0.05494	<input type="checkbox"/>
hr_pc	Medium ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.09000"/>	0.01530	<input type="checkbox"/>
erp_pc	High ▼	Average Response Time	<input style="width: 50px;" type="text" value="0.07000"/>	0.02209	<input type="checkbox"/>

Authorized Actions

Select types of resource allocation actions that may be automatically implemented by QoS Management.

Promote or Demote a Performance Class Consumer Group. Move a CPU between databases within a server pool.

Move a server between server pools.

▼ **Server Pool Directive Override**

2. Specify the new ranks or objective values for the Performance Classes, or enable or disable the Measure Only setting for a Performance Class.
3. When you have finished making your changes, click **OK** to return to the Policy Set Editor wizard.
4. Click **Next** until you reach the end of the wizard. Review your changes, then click **Submit Policy Set**.

Related Topics

- [Monitoring Performance with Oracle Database QoS Management](#)
- [Setting Server Pool Directive Overrides](#)
A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.

9.6.3 Copying a Performance Policy

Instead of creating a new Performance Policy, you can instead copy an existing Performance Policy. A copied Performance Policy is identical to the original Performance Policy. You can then simply rename and edit the copy instead of re-creating all the details in a new Performance Policy.

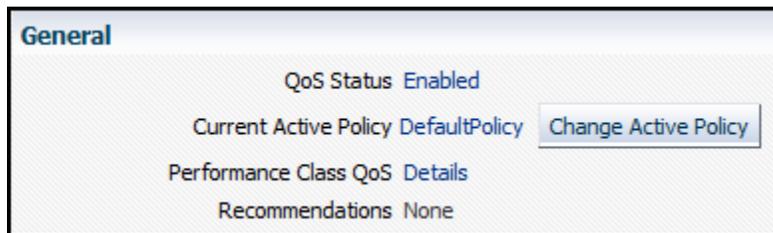
9.6.4 Setting the Current Performance Policy

A Performance Policy is a collection of Performance Objectives, one for each Performance Class, that are in force at the same time. There are multiple ways to modify the active Performance Policy for Oracle Database QoS Management.

- [Changing the Active Performance Policy from the Dashboard](#)
- [Changing the Active Performance Policy from the Policy Set Editor Wizard](#)
- [Changing the Active Performance Policy using a Script](#)

9.6.4.1 Changing the Active Performance Policy from the Dashboard

1. On the Dashboard page, in the General section, click the button **Change Active Policy**.



2. On the Set Policy page, select the Performance Policy you want to use, then click **OK**.

9.6.4.2 Changing the Active Performance Policy from the Policy Set Editor Wizard

You can change the Performance Policy that will be active when you submit the Policy Set to Oracle Database QoS Management.

1. Start the Policy Set Editor wizard.
2. Proceed to the fifth page in the wizard, which is titled Edit Policy Set: Set Policy.
3. Select the Performance Policy you want enforced, and click **Set Policy**.

At the end of the Policy Set Editor wizard, you can review the settings you specified, then click **Submit Policy Set** to configure Oracle Database QoS Management.

9.6.4.3 Changing the Active Performance Policy using a Script

Many Oracle Database QoS Management policies are calendar based. You can switch the active performance policy automatically through a job scheduler such as Enterprise Manager Cloud Control, Task Scheduler or CRON by using a QOSCTL command to set the active policy.

1. Log in to the operating system user as the Clusterware administrator.
2. At the command line, or within a script, use the `qosctl` command with the `-activatepolicy` option. The command must use the following syntax:

```
qosctl qos_admin_username -activatepolicy policy_name
```

If the Performance Policy name includes spaces, then you must enclose the Performance Policy name within double quotes, for example:

```
gosctl qosadmin -activatepolicy "Business Hours"
```

Related Topics

- [QOSCTL Utility Reference](#)
QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.

9.6.5 Deleting a Performance Policy

To delete a Performance Policy, perform the following steps:

1. Start the Policy Set Editor wizard.
2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.
3. Click the **Delete Policy** button to delete a Performance Policy.
4. You then advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

9.6.6 Automatically Implementing Recommendations for a Performance Policy

You can use the appropriate check boxes to specify which of the following actions can be implemented automatically by Oracle Database QoS Management:

- Promote or demote a performance class consumer group
- Move a CPU between databases within a server pool
- Move CPU shares between PDBs
- Move a server between server pools

Authorized Actions

Select types of resource allocation actions that may be automatically implemented by QoS Management.

Promote or Demote a Performance Class Consumer Group. Move CPU Shares between PDBs Move a CPU between databases within a server pool. Move a server between server pools.

If you do not authorize any of these actions, then Oracle Database QoS Management does not implement any changes to the active system until you review the current Recommendations for a Performance Class and click the **Implement** button.

9.6.7 Setting Server Pool Directive Overrides

A server pool directive override gives you the ability to enforce different settings for server pool sizes, or change the importance of server pools.

For example, if you are expecting a surge in demand, such as during an advertised sale period, then you could use a server pool directive override to allocate more

resources to the accounting applications. Server pool directive overrides should be used only when necessary. Instead of using a server pool directive override, you should monitor the system over time and modify the server pool settings as needed. Using a server pool directive override can result in unexpected changes in resource allocations. For example, assume you have server pools named `webapps`, `HR`, and `payroll`. You create a server pool directive override to increase the minimum server pool size for the `payroll` server pool. When the server pool directive override is active, a server could be removed from the `HR` or `webapps` server pool to satisfy the higher minimum server requirement of the `payroll` server pool.

1. Start the Policy Set Editor wizard.
2. Proceed to the fourth page in the wizard, which is titled Edit Policy Set: Performance Policies.
3. Click the **Edit Policy** button to edit a Performance Policy.
The Edit policy page appears.
4. Expand the Server Pool Directive Override section, if necessary.

Server Pool Directive Override								
Server Pool Directive Overrides change the server pool availability properties when the associated policy is in effect. ⓘ								
Server Pool	Server Pool Size						Importance	
	Min		Max		Current	Importance		
	Current Value	Override	Current Value	Override		Current Value	Override	
backoffice	1	<input type="text" value="1"/>	-1	<input type="text" value="-1"/>	2	10	<input type="text" value="10"/>	
Free	0	<input type="text" value="null"/>	-1	<input type="text" value="null"/>	0	0	<input type="text" value="0"/>	
online	1	<input type="text" value="1"/>	-1	<input type="text" value="-1"/>	2	20	<input type="text" value="20"/>	

5. To set server pool directive overrides, perform the following:
 - To override the current value for the minimum number of servers in a server pool, enter a new value in the Min:Override field for that server pool. Valid values are 0 to the maximum number of servers in that server pool.
 - To override the current value for the maximum number of servers in a server pool, enter a new value in the Max:Override field for that server pool. Valid values are from the minimum number of servers in that server pool to the size of the cluster.
 - To override the current value for the Importance of the server pool, enter a new value in the Importance:Override field for that server pool. Valid values are 0 to 1000; higher values indicate greater importance.
6. After you have finished entering the server pool directive override values, click **OK** to implement the changes.
7. Advance to the end of the Edit Policy Set wizard, and click **Submit Policy Set** to make the change permanent.

Related Topics

- [Modifying Server Pool Settings](#)
To modify the server pool settings, you configure a server pool directive override for a Policy Set.

9.7 Reviewing Performance Metrics

You can view a variety of performance metrics for the Oracle Database QoS Management system as a whole, or for individual Performance Classes.

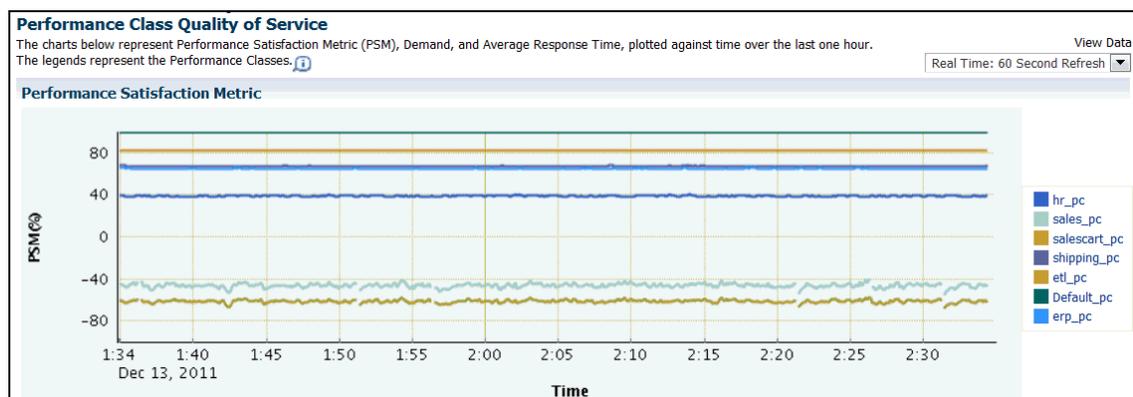
- [Viewing Performance Metrics for All Performance Classes](#)
You can view the performance metrics for your system on the Performance Class Quality of Service page.
- [Viewing Performance Metrics for Individual Performance Classes](#)
After you have configured Oracle Database Quality of Service Management, and a short period of time has passed, you can view the performance metrics for a specific Performance Class.
- [Configuring Alerts for Quality of Service Management Events](#)
It is not convenient or efficient to require constant manual monitoring of the Quality of Service Management Dashboard. Instead you can use the Enterprise Manager Cloud Control notification system for reporting negative Performance Satisfaction Metrics (PSMs) that persist for user-specified times.
- [Viewing the Resource Wait Times Breakdown](#)

9.7.1 Viewing Performance Metrics for All Performance Classes

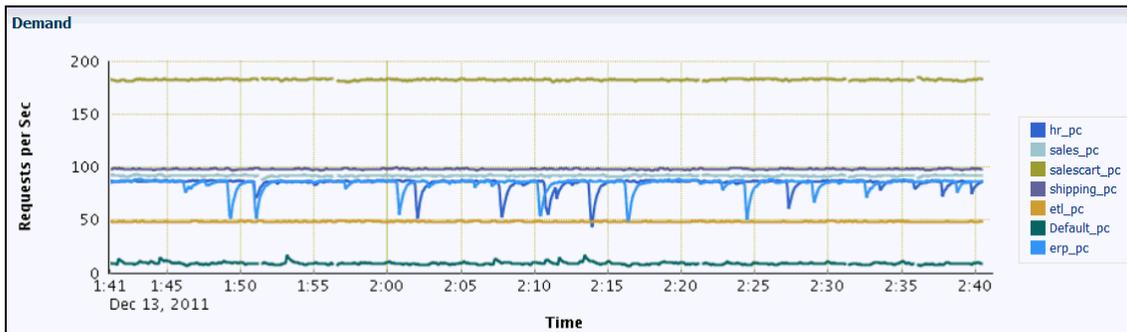
You can view the performance metrics for your system on the Performance Class Quality of Service page.

After you have configured Oracle Database QoS Management, a short period of time is required for Oracle Database QoS Management to gather performance data and evaluate the performance of the system. After this period of time has passed, you can view the performance metrics for your system. To view the current performance metrics, perform the following steps:

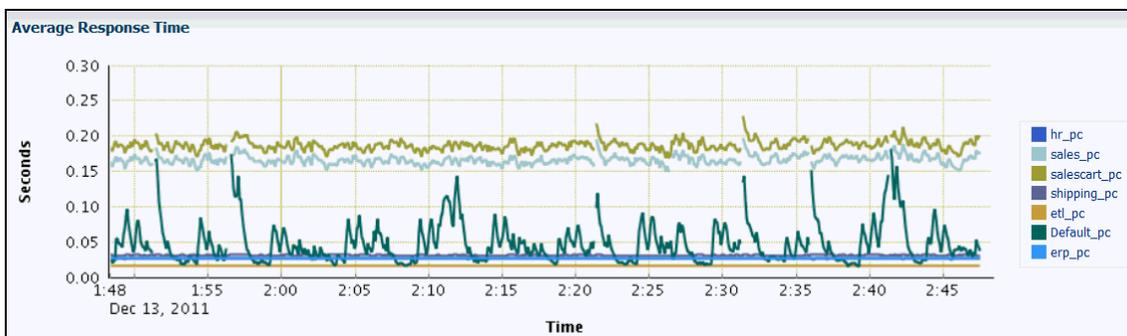
1. Log in to Oracle Enterprise Manager Cloud Control as the cluster administrator. Go to the cluster target page.
2. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then **View Performance Class Quality of Service**.
3. The Performance Class Quality of Service page displays three charts measuring the current performance of each Performance Class that is being monitored:
 - a. The Performance Satisfaction Metric chart



b. The Demand chart



c. The Average Response Time chart



9.7.2 Viewing Performance Metrics for Individual Performance Classes

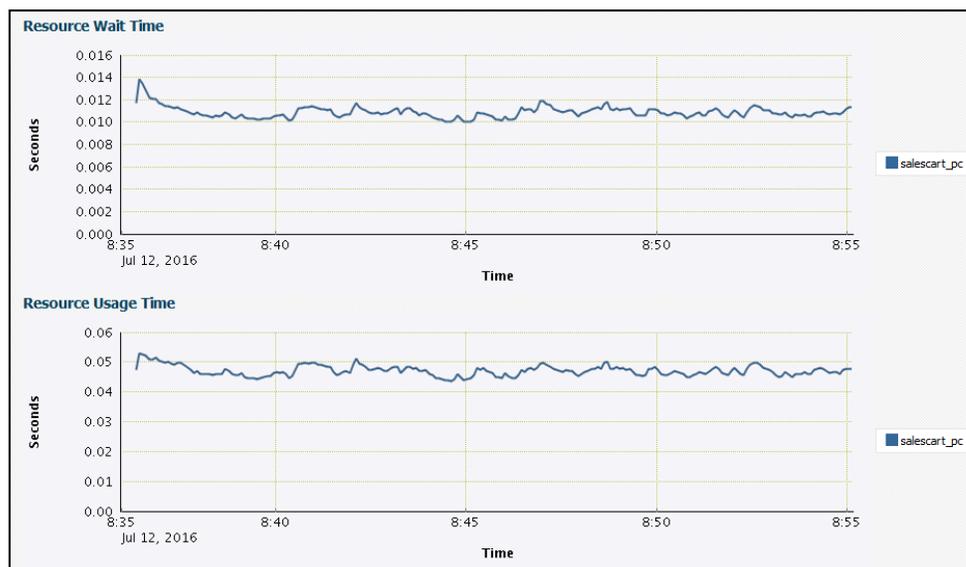
After you have configured Oracle Database Quality of Service Management, and a short period of time has passed, you can view the performance metrics for a specific Performance Class.

To view the current performance metrics for a Performance Class, perform the following steps:

1. Log in to Oracle Enterprise Manager Cloud Control as the cluster administrator.
2. Select the cluster target page that is configured for QoS Management.
3. From the cluster target menu, select **Administration**, then **Quality of Service Management**, then **View Performance Class Quality of Service**.
4. On the right-hand side of any graph, in the legend box, click the link that corresponds to the Performance Class for which you want to view the performance metrics.

When you view the performance metrics for an individual Performance Class, you can see two additional graphs:

- Resource Usage Time
- Resource Wait Time

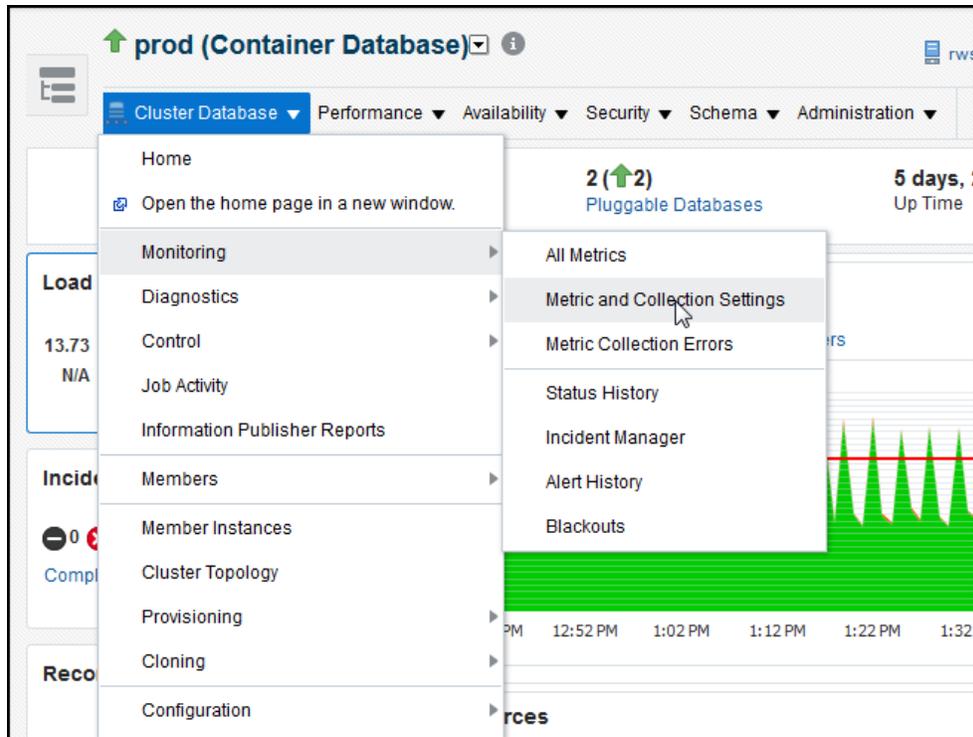
Figure 9-5 Resource Wait Time and Resource Usage Time Charts for a Performance Class

9.7.3 Configuring Alerts for Quality of Service Management Events

It is not convenient or efficient to require constant manual monitoring of the Quality of Service Management Dashboard. Instead you can use the Enterprise Manager Cloud Control notification system for reporting negative Performance Satisfaction Metrics (PSMs) that persist for user-specified times.

Both warning and critical levels can be alerted based upon specified durations for each performance class by setting up alert thresholds and notifications. The alerts are configured against the databases that offer the services being monitored

1. Start Enterprise Manager Cloud Control.
2. Navigate to the database for which you want to configure the alert.
3. From the **Cluster Database** menu select **Monitoring** and then select| **Metric and Collection Settings**.



- Under the **Metrics** tab, edit the metrics in the category **QoS Management – Performance Satisfaction Metrics**.

Monitoring User Expiry				Every 24 Hours	
Account Expiry In Hours	<	72		None	
QoS Management - Performance Satisfaction Metric				Every 5 Minutes	
Negative PSM Duration (seconds)	>			None	
Response				Event-Driven	

For example, if Quality of Service Management uses the database service `sales_svc_pc`, then under Negative PSM Duration (seconds), for that service, you might configure a warning alert when the duration is more than 120 seconds and a critical alert if the duration is more than 180 seconds.

The configured alerts will appear on the Database Home page of Enterprise Manager if a violation is detected.

9.7.4 Viewing the Resource Wait Times Breakdown

At the bottom of the Dashboard is the Resource Wait Times Breakdown table. This table provides breakdown of resource wait times by Performance Class. For each Performance Class, the bottlenecked resource is the one that has the most wait time. This data is used by Oracle Database QoS Management to produce recommendations. You can also use this data to make manual adjustments to your system.

Resource Wait Times Breakdown

This table provides breakdown of resource wait times by Performance Class. For each performance class, the bottlenecked resource is the one that has the most wait time. This data is used by QoS Management to produce Recommendations. The data can also be used to make manual adjustments to the system.

Expand All | Collapse All

Performance Class/Server Pool	CPU (sec)	Global Cache (sec)	IO (sec)	Other (sec)
▼ rwsbi0508-r				
▶ salescart_pc	0.121691	0.000000	0.000000	0.000755
▶ erp_pc	0.000850	0.000000	0.000000	0.000116
▶ sales_pc	0.112963	0.000000	0.000000	0.000214
▶ shipping_pc	0.001970	0.000000	0.000000	0.000095
▶ etl_pc	0.000451	0.000000	0.000000	0.000040
▶ hr_pc	0.000886	0.000000	0.000000	0.000114
▼ Default_pc	0.051329	0.000000	0.000000	0.012530
online	0.051284	0.000000	0.000000	0.012375
backoffice	0.000045	0.000000	0.000000	0.000155

9.8 Creating Administrative Users for Oracle Database QoS Management

Oracle Database QoS Management provides a command line utility named QOSCTL to help you manage users. This utility is installed on each node of the cluster but runs properly only if executed as the Oracle Clusterware administrator user on the same node on which the Oracle Database QoS Management server is running. The correct node to run the utility on can be determined by issuing the following command:

```
srvctl status qosmsserver
```

The QOSCTL utility supports the creation of Oracle Database QoS Management administrative users. The account information is stored in the local `system-jazn-data.xml` file with encrypted credentials. The account information is also stored in the Oracle Clusterware Repository (OCR) to support failover of the Oracle Database QoS Management Server.

- [QOSCTL Utility Reference](#)
 QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.

9.8.1 QOSCTL Utility Reference

QOSCTL is a command-line utility that allows you to perform certain configuration tasks for Oracle Database QoS Management.

Purpose

The QOSCTL utility supports the creation of Oracle Database QoS Management administrative users.

File Path

The `qosctl` executable file is located in the `Grid_home/bin` directory.

Security Requirements

To use the QOSCTL utility, you must be logged in as an Oracle Clusterware administrator user.

Displaying Help for the QOSCTL Utility

To display the help for the `qosctl` utility, use the following command:

```
qosctl -help
```

Syntax

The following code example shows the general format of QOSCTL commands:

```
qosctl qos_admin_user command
```

In place of the `qos_admin_user` argument, you would put the user name of an Oracle Database QoS Management administrative user. Do not include the password. When you have submitted the command for execution, you are prompted for the password associated with the specified `qos_admin_user`.

When configuring the initial accounts for your Oracle Database QoS Management system, the command uses operating system authentication.

Commands

Table 9-1 Summary of Commands for the QOSCTL Utility

Command Syntax	Description
<code>-activatepolicy policy_name</code>	Sets the current policy. This command enables you to change policies through a CRON or scheduling job. Policy names that include spaces must be encapsulated in double-quotes.
<code>-updatepolicyset</code>	Discovers and adds new databases with their respective services, and adds performance classes for each service to the existing policy set.
<code>-adduser username</code>	Adds the specified user and prompts for a password. This user is automatically granted the role required to use the Oracle Database QoS Management application and execute any of these commands.

Table 9-1 (Cont.) Summary of Commands for the QOSCTL Utility

Command Syntax	Description
-listusers	Lists the users that are authorized to run the Oracle Database QoS Management application
-remuser <i>username</i>	Removes the specified user account and all associated permissions Note: This command is not recoverable. Use caution, because you can delete your own account.
-setpasswd <i>username</i>	Updates the password of a specified user. The QOSCTL utility prompts you for the current password for the user. You must supply the correct value for the old password to change the password for a user. If the password for a user has been forgotten or lost, then you should remove the user account and create a new account for the user, with a new password.
-help	Displays the syntax for QOSCTL commands.

9.9 Editing the Resource Plan for Oracle Database QoS Management

You can perform limited editing of the Resource Manager plans used by Oracle Database Quality of Service Management.

Oracle Database QoS Management activates a resource plan named `APPQOS_PLAN`, which is a complex, multilevel resource plan. Oracle Database QoS Management also creates consumer groups that represent Performance Classes and resource plan directives for each consumer group.

You can modify some of the sections of the `APPQOS_PLAN`, or `ORA$QOS_PLAN` and `ORA$QOS_CDB_PLAN` for Multitenant databases, but only as specified in the following table:

Plan Component	Modifiable?	Description
General	DO NOT EDIT	This section must not be edited as it is key to the QoS Management modeling and recommendation engine. If you modify this section, the change will be detected. In that case, Oracle Enterprise Manager Cloud Control reports a database error on the QoS Management dashboard.
Parallelism	Limited edits allowed	This section can be edited however all values set for the <code>ORA\$APPQOS*</code> consumer groups must be the same within each column due to the reason stated above.
Thresholds	Limited edits allowed	This section can be edited however all values set for the <code>ORA\$APPQOS*</code> consumer groups must be the same within each column due to the reason stated above. In addition, none of the "Switch to *" actions should be selected for these groups.
Idle Time	Limited edits allowed	This section can be edited however all values set for the <code>ORA\$APPQOS*</code> consumer groups must be the same within each column due to the reason stated above.

Related Topics

- [Oracle Database Resource Manager](#)
Oracle Database QoS Management uses Oracle Database Resource Manager to manage allocate resources to performance classes.
- *Oracle Database Administrator's Guide*
- *Oracle Database PL/SQL Packages and Types Reference*

10

Troubleshooting Oracle Database QoS Management

This chapter describes some problems you might encounter when using Oracle Database QoS Management and how you can resolve them. This chapter also describes how to locate the trace or log files for Oracle Database QoS Management.

- [Common Problems](#)
- [Locating Log or Trace Files](#)
- [Enabling Tracing](#)
Tracing can assist in troubleshooting problems with Oracle Database Quality of Service Management.

10.1 Common Problems

After the initial configuration, Oracle Database Quality of Service Management is an automated system. As a result, most of the problems you might encounter are related to configuring Oracle Database QoS Management. The following sections illustrate the most common problems, and how to resolve them:

- [Cannot Enable Oracle Database Quality of Service Management](#)
- [Cannot Enable Oracle Database QoS Management for a Database](#)
- [Oracle Database Resource Manager Not Enabled and Resource Plan Errors](#)
Oracle Database Quality of Service (QoS) Management requires Database Resource Manager to use a specific resource plan.
- [Do Not Have Access to a Server Pool](#)
- [Server Pool Is Marked As Unmanageable](#)
- [Metrics Are Missing For a Performance Class](#)
- [Oracle Database QoS Management is not Generating Recommendations](#)
- [Recently Added Server was Placed in the Wrong Server Pool](#)
Servers are moved within a cluster by Oracle Clusterware, or as directed by an administrator.
- [RMI Port Conflict Detected](#)

10.1.1 Cannot Enable Oracle Database Quality of Service Management

Before you can enable Oracle Database QoS Management within a cluster, you must first create a Policy Set.

Related Topics

- [Create an Initial Policy Set](#)

10.1.2 Cannot Enable Oracle Database QoS Management for a Database

For a database to be managed by Oracle Database QoS Management, the database must be compliant, enabled, and Oracle Database QoS Management must be able to access APPQOSSYS database user:

- A *compliant* database is an Oracle RAC database that is running Oracle Database release 11.2.0.2 or greater.
- For a database to be *enabled*, Oracle Database QoS Management must be able to connect to the database. The connection is configured when you select the **Enable Quality of Service Management** link in the Oracle Enterprise Manager Cloud Control. When you select this link, you are prompted for the cluster credentials and the password for the APPQOSSYS account in the database. By default the APPQOSSYS account is expired. When you submit a password, the account is unlocked.
- If the password for the APPQOSSYS user is changed through other methods or the account is locked, then Oracle Database QoS Management is disabled for this database until this condition is corrected by selecting the **Enable Quality of Service Management** link again.

Related Topics

- [Enabling Oracle Database QoS Management for a Cluster](#)

10.1.3 Oracle Database Resource Manager Not Enabled and Resource Plan Errors

Oracle Database Quality of Service (QoS) Management requires Database Resource Manager to use a specific resource plan.

Oracle Database QoS Management installs a special Oracle Database Resource Manager plan, `APPQOS_PLAN`, whenever a database is enabled for management by Oracle Database QoS Management. Oracle Database QoS Management requires this resource plan to move Performance Classes to different levels of CPU scheduling. No other plan can be active while Oracle Database QoS Management is enabled on this database. If a resource plan is not enabled for the Oracle RAC database, then an error results when trying to enable the database for management by Oracle Database QoS Management. After a resource plan is enabled, then enabling the database for management by Oracle Database QoS Management succeeds.

To check that the required `APPQOS_PLAN` is active on the target database, connect using SQL*Plus and issue the following statement:

```
SQL> show parameter resource_manager_plan
```

You should see output similar to the following:

NAME	TYPE	VALUE
resource_manager_plan	string	FORCE:APPQOS_PLAN

If you are querying a Multitenant database, then the output is similar to the following:

NAME	TYPE	VALUE
resource_manager_plan	string	FORCE:ORA\$QOS_CDB_PLAN

10.1.4 Do Not Have Access to a Server Pool

Server pools can be managed separately from the database by configuring special operating system groups. By default, the user that installed Oracle Grid Infrastructure for a cluster can perform operations on server pools. If you use a separate operating system for the Oracle Database installation, then execute permissions on a server pool must be granted to the database software owner before a database can be deployed in that server pool. To grant this permission, you must use the CRSCTL utility to modify the server pool ACL attribute. A database administrator can also create server pools using Server Control (SRVCTL) or Oracle Enterprise Manager.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

10.1.5 Server Pool Is Marked As Unmanageable

A server pool is marked as unmanageable if Oracle Database QoS Management is not able to properly measure or predict the performance of Performance Classes deployed in that server pool. A server pool can be considered unmanageable under the following conditions:

1. The servers in the server pool have different physical CPU counts.
2. The CPU count for every database instance could not be retrieved.
3. The sum of the configured CPU counts for all database instances on a server is greater than its physical CPU count.
4. Singleton services are deployed in a server pool with a maximum size larger than one.
5. Oracle Database QoS Management is unable to collect the metrics for all Performance Classes, or the metrics collected do not contain valid data.
6. The server pool has a database that is not enabled for Oracle Database QoS Management.

If a server is added to a server pool, then Oracle starts up instances and services on the new server for the policy-managed databases in the server pool. When an instance is started on the new server, Oracle checks the SPFILE of the existing instances for the `CPU_COUNT` setting, and uses this value for the new instance. By default, if there is no setting for `CPU_COUNT` in the SPFILE, then the instances on the new server will be started with `CPU_COUNT` set to the number of physical CPUs of the server. This is not a supported configuration and will result in the server pool being marked as unmanageable. Also, if you modify the `CPU_COUNT` parameter but do not store the change in the SPFILE, then the `CPU_COUNT` parameter might be set to the wrong value on the newly started instance resulting in a configuration violation.

Make sure the databases and database instances that you want to be managed by Oracle Database QoS Management conform to the requirements documented in "[Supported Database Configurations](#)".

10.1.6 Metrics Are Missing For a Performance Class

Metrics are captured by Oracle Database QoS Management by querying the managed database instances in the cluster. Metrics are not displayed under the following circumstances:

1. The database requests (work being done) do not match the classifiers defined for a specific Performance Class and "No demand" is displayed.
2. A more general classifier of a Performance Class is evaluated first. As a result, the Performance Class with the more specific classifier is shown as having "No demand". For example, if the `sales_pc` Performance Class uses the classifier `service=Sales` and the `sales_search` Performance Class uses the classifier `(service=Sales and action=Search)`, then, if the `sales_pc` Performance Class was listed first in the Edit Policy Set Performance Class Order screen, then any work request that uses the `sales` service would be placed in the `sales_pc` Performance Class, including those that are performing the `Search` action. The `sales_search` Performance Class does not generate any metrics and "No demand" is displayed for that Performance Class.
3. The database instance is over-utilized and the metrics query reaches the time out limit and "Incomplete data" is displayed.
4. The database is not producing mutually consistent data for the Performance Class and "Nonconforming data" is displayed. For example a Performance Class that is classifying database requests that exceed one second will cause this response.
5. The metrics collected through the query do not pass the sanity verification checks performed by Oracle Database QoS Management and "Nonconforming data" is displayed.

10.1.7 Oracle Database QoS Management is not Generating Recommendations

Recommendations are generated once a minute when logged into the Oracle Database QoS Management Dashboard (the Dashboard). Recommendations do not appear on the Dashboard in the following cases:

1. Oracle Database QoS Management is disabled.
2. An action is in progress implementing a recommendation.
3. During the first minute of enabling Oracle Database QoS Management, or when submitting a Policy Set or activating a Policy Set.

10.1.8 Recently Added Server was Placed in the Wrong Server Pool

Servers are moved within a cluster by Oracle Clusterware, or as directed by an administrator.

When a new server joins the cluster, Oracle Clusterware places the server in a server pool according to the placement algorithm and the state of the server pool attributes of Min, Max and Importance. See the *Oracle Clusterware Administration and Deployment Guide* for a complete description of this placement process.

Related Topics

- *Oracle Clusterware Administration and Deployment Guide*

10.1.9 RMI Port Conflict Detected

When you install Oracle Grid Infrastructure for a cluster, the port for the `qosmsserver` resource might be set to 23792. This can cause a port conflict with Java Remote Method Invocation (RMI). You should set the `qosmsserver` resource to use an available port, then restart the `qosmsserver` resource using the following commands:

```

srvctl modify qosmsserver -rmiport port
srvctl stop qosmsserver
srvctl start qosmsserver

```

10.2 Locating Log or Trace Files

Use the following table to locate a specific log or trace file for Oracle Database QoS Management.

Log File	Location
Oracle Database QoS Management server operations log file	<i>Oracle_Base/crsdata/ node_name/qos/logs/dbwlm/ auditing/log0.xml</i>
Oracle Database QoS Management server trace log file	<i>Oracle_Base/crsdata/ node_name/qos/logs/dbwlm/ logging/log0.xml</i>
Standard out log file and the trace file for the Server Manager (SRVM) component of Oracle Clusterware	<i>Oracle_Base/crsdata/ node_name/qos/logs/ qosmsserver_stdout.trc.n</i>
<code>qosmsserver</code> standard error log file	<i>Oracle_Base/crsdata/ node_name/qos/logs/ catalina.timestamp.log</i>
Start and stop log file for the DBWLM component of Oracle Database QoS Management	<i>Oracle_Base/crsdata/ node_name/qos/logs/ catalina.timestamp.log</i>

`log0.xml` is the name of the most recent log file. Older log files are named in increasing sequential order.

10.3 Enabling Tracing

Tracing can assist in troubleshooting problems with Oracle Database Quality of Service Management.

A default level of tracing is set at installation time. Finer-grained tracing can be enabled under the direction of Oracle Support Services.

Part IV

Automatic Problem Solving

Some situations can be automatically resolved with tools in the Autonomous Health Framework.

- [Resolving Memory Stress](#)
Memory Guard continuously monitors and ensures the availability of cluster nodes by preventing the nodes from being evicted when the nodes are stressed due to lack of memory.
- [Resolving Database and Database Instance Hangs](#)
Hang Manager preserves the database performance by resolving hangs and keeping the resources available.

11

Resolving Memory Stress

Memory Guard continuously monitors and ensures the availability of cluster nodes by preventing the nodes from being evicted when the nodes are stressed due to lack of memory.

- [Overview of Memory Guard](#)
Memory Guard automatically monitors cluster nodes to prevent node stress caused by the lack of memory.
- [Memory Guard Architecture](#)
Memory Guard is implemented as a daemon running as an MBean in a J2EE container managed by Cluster Ready Services (CRS).
- [Enabling Memory Guard in Oracle Real Application Clusters \(Oracle RAC\) Environment](#)
Memory Guard is automatically enabled when you install Oracle Grid Infrastructure for an Oracle Real Application Clusters (Oracle RAC) or an Oracle RAC One Node database.
- [Use of Memory Guard in Oracle Real Application Clusters \(Oracle RAC\) Deployment](#)
Memory Guard autonomously detects and monitors Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node databases when they are open.

Related Topics

- [Introduction to Memory Guard](#)
Memory Guard is an Oracle Real Application Clusters (Oracle RAC) environment feature to monitor the cluster nodes to prevent node stress caused by the lack of memory.

11.1 Overview of Memory Guard

Memory Guard automatically monitors cluster nodes to prevent node stress caused by the lack of memory.

Memory Guard autonomously collects metrics on memory usage for every node in an Oracle Real Application Clusters (Oracle RAC) environment. Memory Guard gets the information from Cluster Health Monitor. If Memory Guard determines that a node has insufficient memory, then Memory Guard performs the following actions:

- Prevents new database sessions from being created on the afflicted node
- Stops all CRS-managed services transactionally on the node, allowing the existing workload on the node to complete and free their memory

When Memory Guard determines that the memory stress has been relieved, it restores connectivity to the node, allowing new sessions to be created on that node.

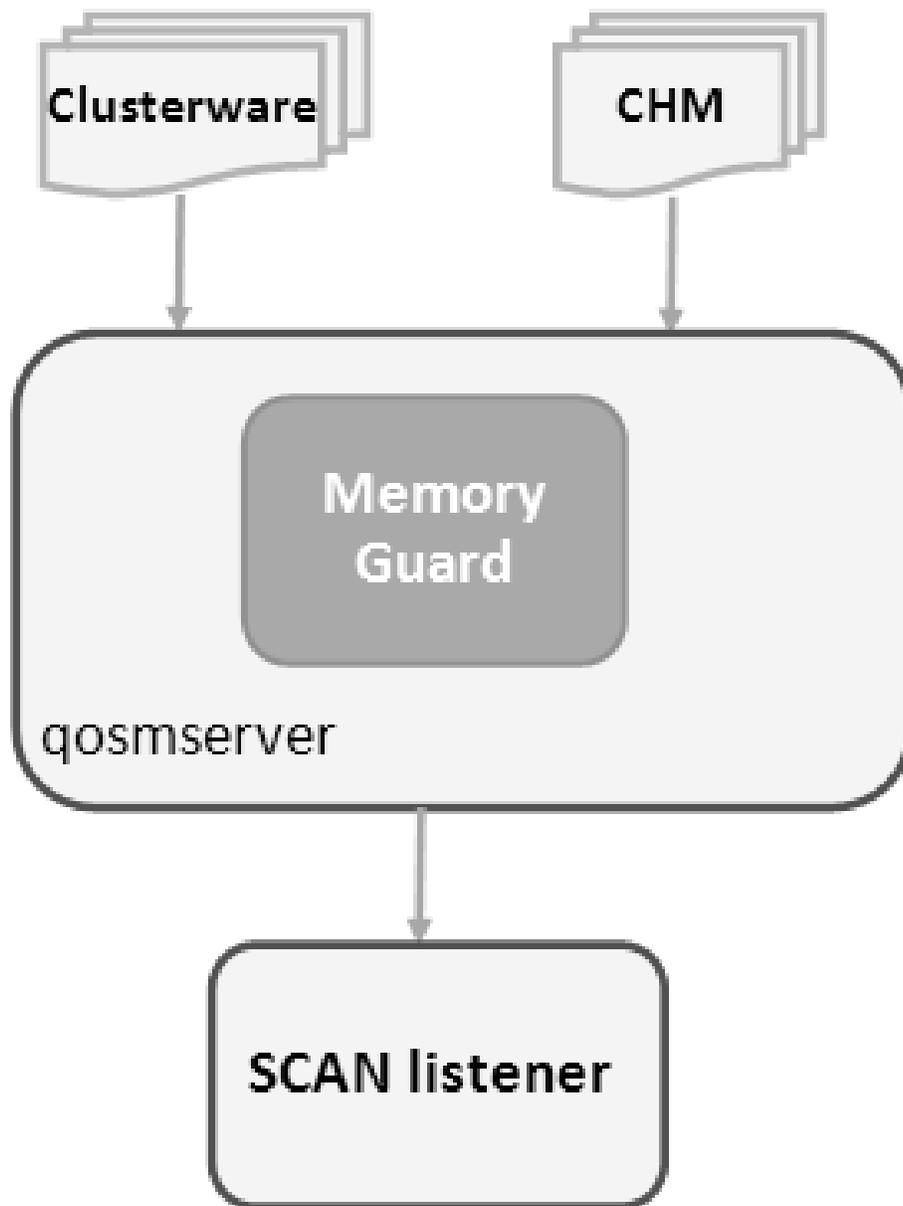
Running out of memory can result in failed transactions or, in extreme cases, a restart of the node resulting in the loss of availability and resources for your applications.

11.2 Memory Guard Architecture

Memory Guard is implemented as a daemon running as an MBean in a J2EE container managed by Cluster Ready Services (CRS).

Memory Guard is hosted on the `qosmserver` resource that runs on any cluster node for high availability.

Figure 11-1 Memory Guard Architecture



Cluster Health Monitor sends a metrics stream to Memory Guard that provides real-time information about memory resources for the cluster nodes. This information includes the following:

- Amount of available memory
- Amount of memory currently in use

After getting memory resource information, Memory Guard collects the cluster topology from Oracle Clusterware. Memory Guard uses cluster topology and memory metrics to identify database nodes that have memory stress. Memory is considered stressed when the free memory is less than a certain threshold.

Memory Guard then stops the database services managed by Oracle Clusterware on the stressed node transactionally. Memory Guard relieves the memory stress without affecting already running sessions and their associated transactions. After completion, the memory used by these processes starts freeing up and adding to the pool of the available memory on the node. When Memory Guard detects that the amount of available memory is more than the threshold, it restarts the services on the affected node.

While a service is stopped on a stressed node, the listener redirects new connections for that service to other nodes that provide the same service for non-singleton database instances. However, for the policy-managed databases, the last instance of a service is not stopped to ensure availability.

 **Note:**

Memory Guard can start or stop the services for databases in the **Open** state. Memory Guard does not manage the default database service and does not act while upgrading or downgrading a database.

11.3 Enabling Memory Guard in Oracle Real Application Clusters (Oracle RAC) Environment

Memory Guard is automatically enabled when you install Oracle Grid Infrastructure for an Oracle Real Application Clusters (Oracle RAC) or an Oracle RAC One Node database.

Run the `srvctl` command to query the status of Memory Guard as follows:

```
srvctl status qosmsserver
```

Example 11-1 Verifying that Memory Guard is Running on a Node

The following example shows sample output of the status of Memory Guard on `qosmsserver`.

```
$ srvctl status qosmsserver
QoS Management Server is enabled.
QoS Management Server is running on node nodeABC
```

11.4 Use of Memory Guard in Oracle Real Application Clusters (Oracle RAC) Deployment

Memory Guard autonomously detects and monitors Oracle Real Application Clusters (Oracle RAC) or Oracle RAC One Node databases when they are open.

Memory Guard sends alert notifications when Memory Guard detects memory stress on a database node. You can find Memory Guard alerts in audit logs at `$ORACLE_BASE/crsdata/node name/qos/logs/dbwlm/auditing`.

Example 11-2 Memory Guard Alert Notifications

The following example shows a Memory Guard log file when the services were stopped due to memory stress.

```
<MESSAGE>
<HEADER>
<TSTZ_ORIGINATING>2016-07-28T16:11:03.701Z</TSTZ_ORIGINATING>
<COMPONENT_ID>wlm</COMPONENT_ID>
<MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
<MSG_LEVEL>1</MSG_LEVEL>
<HOST_ID>hostABC</HOST_ID>
<HOST_NWADDR>11.111.1.111</HOST_NWADDR>
<MODULE_ID>gomlogger</MODULE_ID>
<THREAD_ID>26</THREAD_ID>
<USER_ID>userABC</USER_ID>
<SUPPL_ATTRS>
<ATTR NAME="DBWLM_OPERATION_USER_ID">userABC</ATTR>
<ATTR NAME="DBWLM_THREAD_NAME">MPA Task Thread 1469722257648</ATTR>
</SUPPL_ATTRS>
</HEADER>
<PAYLOAD>
<MSG_TEXT>Server Pool Generic has violation risk level RED.</MSG_TEXT>
</PAYLOAD>
</MESSAGE>
<MESSAGE>
<HEADER>
<TSTZ_ORIGINATING>2016-07-28T16:11:03.701Z</TSTZ_ORIGINATING>
<COMPONENT_ID>wlm</COMPONENT_ID>
<MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
<MSG_LEVEL>1</MSG_LEVEL>
<HOST_ID>hostABC</HOST_ID>
<HOST_NWADDR>11.111.1.111</HOST_NWADDR>
<MODULE_ID>gomlogger</MODULE_ID>
<THREAD_ID>26</THREAD_ID>
<USER_ID>userABC</USER_ID>
<SUPPL_ATTRS>
<ATTR NAME="DBWLM_OPERATION_USER_ID">userABC</ATTR>
<ATTR NAME="DBWLM_THREAD_NAME">MPA Task Thread 1469722257648</ATTR>
</SUPPL_ATTRS>
</HEADER>
<PAYLOAD>
MSG_TEXT>Server userABC-hostABC-0 has violation risk level RED. New connection
requests will no longer be accepted.</MSG_TEXT>
</PAYLOAD>
</MESSAGE>
```

The following example shows a Memory Guard log file when the services were restarted after relieving the memory stress.

```

<MESSAGE>
<HEADER>
<TSTZ_ORIGINATING>2016-07-28T16:11:07.674Z</TSTZ_ORIGINATING>
<COMPONENT_ID>wlm</COMPONENT_ID>
<MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
<MSG_LEVEL>1</MSG_LEVEL>
<HOST_ID>hostABC</HOST_ID>
<HOST_NWADDR>11.111.1.111</HOST_NWADDR>
<MODULE_ID>gomlogger</MODULE_ID>
<THREAD_ID>26</THREAD_ID>
<USER_ID>userABC</USER_ID>
<SUPPL_ATTRS>
<ATTR NAME="DBWLM_OPERATION_USER_ID">userABC</ATTR>
<ATTR NAME="DBWLM_THREAD_NAME">MPA Task Thread 1469722257648</ATTR>
</SUPPL_ATTRS>
</HEADER>
<PAYLOAD>
<MSG_TEXT>Memory pressure in Server Pool Generic has returned to normal.</MSG_TEXT>
</PAYLOAD>
</MESSAGE>
<MESSAGE>
<HEADER>
<TSTZ_ORIGINATING>2016-07-28T16:11:07.674Z</TSTZ_ORIGINATING>
<COMPONENT_ID>wlm</COMPONENT_ID>
<MSG_TYPE TYPE="NOTIFICATION"></MSG_TYPE>
<MSG_LEVEL>1</MSG_LEVEL>
<HOST_ID>hostABC</HOST_ID>
<HOST_NWADDR>11.111.1.111</HOST_NWADDR>
<MODULE_ID>gomlogger</MODULE_ID>
<THREAD_ID>26</THREAD_ID>
<USER_ID>userABC</USER_ID>
<SUPPL_ATTRS>
<ATTR NAME="DBWLM_OPERATION_USER_ID">userABC</ATTR>
<ATTR NAME="DBWLM_THREAD_NAME">MPA Task Thread 1469722257648</ATTR>
</SUPPL_ATTRS>
</HEADER>
<PAYLOAD>
<MSG_TEXT>Memory pressure in server userABC-hostABC-0 has returned to normal. New
connection requests are now accepted.</MSG_TEXT>
</PAYLOAD>
</MESSAGE>
<MESSAGE>

```

12

Resolving Database and Database Instance Hangs

Hang Manager preserves the database performance by resolving hangs and keeping the resources available.

- [Hang Manager Architecture](#)
Hang Manager autonomously runs as a `DIA0` task within the database.
- [Optional Configuration for Hang Manager](#)
You can adjust the sensitivity, and control the size and number of the log files used by Hang Manager.
- [Hang Manager Diagnostics and Logging](#)
Hang Manager autonomously resolves hangs and continuously logs the resolutions in the database alert logs and the diagnostics in the trace files.

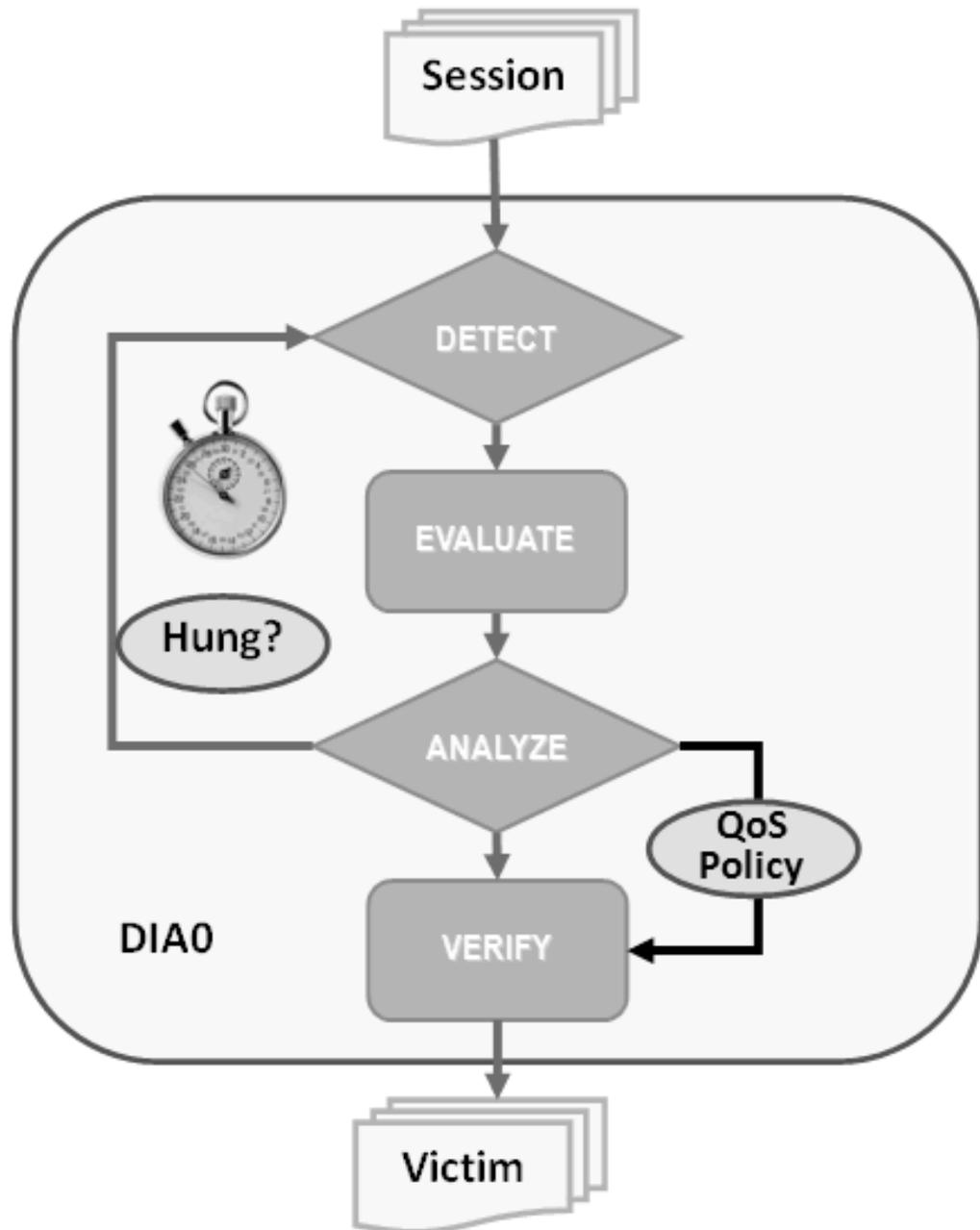
Related Topics

- [Introduction to Hang Manager](#)
Hang Manager is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves hangs and keeps the resources available.

12.1 Hang Manager Architecture

Hang Manager autonomously runs as a `DIA0` task within the database.

Figure 12-1 Hang Manager Architecture



Hang Manager works in the following three phases:

- **Detect:** In this phase, Hang Manager collects the data on all the nodes and detects the sessions that are waiting for the resources held by another session.
- **Analyze:** In this phase, Hang Manager analyzes the sessions detected in the **Detect** phase to determine if the sessions are part of a potential hang. If the sessions are suspected as hung, Hang Manager then waits for a certain threshold time period to ensure that the sessions are hung.

- **Verify:** In this phase, after the threshold time period is up, Hang Manager verifies that the sessions are hung and selects a victim session. The victim session is the session that is causing the hang.

After the victim session is selected, Hang Manager applies hang resolution methods on the victim session. If the chain of sessions or the hang resolves automatically, then Hang Manager does not apply hang resolution methods. However, if the hang does not resolve by itself, then Hang Manager resolves the hang by terminating the victim session. If terminating the session fails, then Hang Manager terminates the process of the session. This entire process is autonomous and does not block resources for a long period and does not affect the performance.

Hang Manager also considers Oracle Database QoS Management policies, performance classes, and ranks that you use to maintain performance objectives.

For example, if a high rank session is included in the chain of hung sessions, then Hang Manager expedites the termination of the victim session. Termination of the victim session prevents the high rank session from waiting too long and helps to maintain performance objective of the high rank session.

12.2 Optional Configuration for Hang Manager

You can adjust the sensitivity, and control the size and number of the log files used by Hang Manager.

Sensitivity

If Hang Manager detects a hang, then Hang Manager waits for a certain threshold time period to ensure that the sessions are hung. Change threshold time period by using `DBMS_HANG_MANAGER` to set the `sensitivity` parameter to either `Normal` or `High`. If the `sensitivity` parameter is set to `Normal`, then Hang Manager waits for the default time period. However, if the sensitivity is set to `High`, then the time period is reduced by 50%.

By default, the `sensitivity` parameter is set to `Normal`. To set Hang Manager sensitivity, run the following commands in SQL*Plus as `SYS` user:

- To set the `sensitivity` parameter to `Normal`:

```
exec dbms_hang_manager.set(dbms_hang_manager.sensitivity,  
dbms_hang_manager.sensitivity_normal);
```

- To set the `sensitivity` parameter to `High`:

```
exec dbms_hang_manager.set(dbms_hang_manager.sensitivity,  
dbms_hang_manager.sensitivity_high);
```

Size of the Trace Log File

The Hang Manager logs detailed diagnostics of the hangs in the trace files with `_base_` in the file name. Change the size of the trace files in bytes with the `base_file_size_limit` parameter. Run the following command in SQL*Plus, for example, to set the trace file size limit to 100 MB:

```
exec dbms_hang_manager.set(dbms_hang_manager.base_file_size_limit, 104857600);
```

Number of Trace Log Files

The base Hang Manager trace files are part of a trace file set. Change the number of trace files in trace file set with the `base_file_set_count` parameter. Run the following command in SQL*Plus, for example, to set the number of trace files in trace file set to 6:

```
exec dbms_hang_manager.set(dbms_hang_manager.base_file_set_count,6);
```

By default, `base_file_set_count` parameter is set to 5.

12.3 Hang Manager Diagnostics and Logging

Hang Manager autonomously resolves hangs and continuously logs the resolutions in the database alert logs and the diagnostics in the trace files.

Hang Manager logs the resolutions in the database alert logs as Automatic Diagnostic Repository (ADR) incidents with incident code `ORA-32701`.

You also get detailed diagnostics about the hang detection in the trace files. Trace files and alert logs have file names starting with `database_instance_dia0_`.

- The trace files are stored in the `$ ADR_BASE/diag/rdbms/database name/database instance/incident/incdir_XXXXXX` directory
- The alert logs are stored in the `$ ADR_BASE/diag/rdbms/database name/database instance/trace` directory

Example 12-1 Hang Manager Trace File for a Local Instance

This example shows an example of the output you see for Hang Manager for the local database instance

```
Trace Log File ../oracle/log/diag/rdbms/hm1/hm1/incident/incdir_111/
hm1_dia0_11111_1111.trc
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
...
*** 2016-07-16T12:39:02.715475-07:00
HM: Hang Statistics - only statistics with non-zero values are listed

      current number of active sessions 3
      current number of hung sessions 1
instance health (in terms of hung sessions) 66.67%
      number of cluster-wide active sessions 9
      number of cluster-wide hung sessions 5
      cluster health (in terms of hung sessions) 44.45%

*** 2016-07-16T12:39:02.715681-07:00
Resolvable Hangs in the System

      Root      Chain Total      Hang
Hang Hang      Inst Root #hung #hung Hang  Hang Resolution
      ID Type Status  Num  Sess  Sess  Sess  Conf  Span  Action
-----
      1 HANG RSLNPEND  3   44   3   5   HIGH GLOBAL Terminate Process
Hang Resolution Reason: Although hangs of this root type are typically
self-resolving, the previously ignored hang was automatically resolved.

kjzshngtbltmp: Hang's QoS Policy and Multiplier Checksum 0x0
```

Inst Num	Sess ID	Ser Num	Proc OSPID	Wait Name	Event
1	111	1234	34567	FG gc	buffer busy acquire
1	22	12345	34568	FG gc	current request
3	44	23456	34569	FG	not in wait

Example 12-2 Error Message in the Alert Log Indicating a Hung Session

This example shows an example of a Hang Manager alert log on the master instance

```
2016-07-16T12:39:02.616573-07:00
Errors in file .../oracle/log/diag/rdbms/hml/hml/trace/hml_dia0_11111.trc
(incident=11111):
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hml/hml/incident/incdir_11111/
hml_dia0_11111_11111.trc
2016-07-16T12:39:02.674061-07:00
DIA0 requesting termination of session sid:44 with serial # 23456 (ospid:34569) on
instance 3
    due to a GLOBAL, HIGH confidence hang with ID=1.
    Hang Resolution Reason: Although hangs of this root type are typically
    self-resolving, the previously ignored hang was automatically resolved.
DIA0: Examine the alert log on instance 3 for session termination status of hang
with ID=1.
```

Example 12-3 Error Message in the Alert Log Showing a Session Hang Resolved by Hang Manager

This example shows an example of a Hang Manager alert log on the local instance for resolved hangs

```
2016-07-16T12:39:02.707822-07:00
Errors in file .../oracle/log/diag/rdbms/hml/hml1/trace/hml1_dia0_11111.trc
(incident=169):
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hml/hml1/incident/incdir_169/
hml1_dia0_30676_1169.trc
2016-07-16T12:39:05.086593-07:00
DIA0 terminating blocker (ospid: 30872 sid: 44 ser#: 23456) of hang with ID = 1
    requested by master DIA0 process on instance 1
    Hang Resolution Reason: Although hangs of this root type are typically
    self-resolving, the previously ignored hang was automatically resolved.
    by terminating session sid:44 with serial # 23456 (ospid:34569)
...
DIA0 successfully terminated session sid:44 with serial # 23456 (ospid:34569) with
status 0.
```

Part V

Collecting Diagnostic Data and Triaging, Diagnosing, and Resolving Issues

Use the tools available with Autonomous Health Framework to troubleshoot errors.

- [Oracle Trace File Analyzer](#)
Oracle Trace File Analyzer helps you collect and analyze diagnostic data.
- [Getting Started with Oracle Trace File Analyzer](#)
This section explains how to install Oracle Trace File Analyzer on different operating systems.
- [Automatic Diagnostic Collections](#)
Oracle Trace File Analyzer monitors your logs for significant problems, such as internal errors like `ORA-00600`, or node evictions.
- [On-demand Analysis and Diagnostic Collection](#)
Run Oracle Trace File Analyzer on demand using `tfactl` command-line tool.
- [Maintaining Oracle Trace File Analyzer to the Latest Version](#)
Oracle releases a new version of Oracle Trace File Analyzer approximately every three months.
- [Performing Custom Collections](#)
Use the custom collection options to change the diagnostic collections from the default.
- [Managing and Configuring Oracle Trace File Analyzer](#)
This section helps you manage Oracle Trace File Analyzer daemon, diagnostic collections, and the collection repository.
- [Oracle Trace File Analyzer Service](#)
Oracle Trace File Analyzer Service is deployed as part of Domain Service Cluster (DSC) setup.
- [Managing Oracle Database and Oracle Grid Infrastructure Diagnostic Data](#)
This section enables you to manage Oracle Database and Oracle Grid Infrastructure diagnostic data and disk usage snapshots.
- [Troubleshooting Oracle Trace File Analyzer](#)
This section helps you diagnose and remediate Oracle Trace File Analyzer issues.

Oracle Trace File Analyzer

Oracle Trace File Analyzer helps you collect and analyze diagnostic data.

As a DBA, you are expected to do more work with fewer resources all the time. You are under pressure to keep the mission-critical applications up and running. When something goes wrong, everyone looks to you to understand what went wrong and how to fix it.

It is not always easy. You have to run the right tools at the right time. If you're using Oracle Clusterware, then you also have to collect diagnostic data from all the database nodes. Collecting this data can require you to use tools that you rarely use. Needless to say, each tool has its own syntax.

The amount of data you collect can be huge. Only a fraction of the data that you collect is useful, but how can you know which part is relevant? You must collect it all, quickly, before the data is overwritten. In the meantime, you have still got a problem that costs your company time and money.

Oracle Trace File Analyzer enables you to collect diagnostic data. Collecting diagnostic data is a crucial step to resolving problems that occur with your Oracle Database.

Oracle Trace File Analyzer monitors your logs for significant problems that potentially impact your service. Oracle Trace File Analyzer also automatically collects relevant diagnostics when it detects any potential problems.

Oracle Trace File Analyzer can identify the relevant information in log files. It trims log files to just the parts that are necessary to resolve an issue. Oracle Trace File Analyzer also collects data across cluster nodes and consolidates everything in one place.

Using important database diagnostic tools is easy with Oracle Trace File Analyzer. Oracle Trace File Analyzer hides the complexity by providing a single interface and syntax for them all.

Related Topics

- [Introduction to Oracle Trace File Analyzer](#)

Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.

14

Getting Started with Oracle Trace File Analyzer

This section explains how to install Oracle Trace File Analyzer on different operating systems.

- [Supported Environments](#)
You can use Oracle Trace File Analyzer with all supported versions of Oracle Database and Oracle Clusterware.
- [Installing Oracle Trace File Analyzer on Linux or UNIX as root User in Daemon Mode](#)
To obtain the fullest capabilities of Oracle Trace File Analyzer, install it as `root`.
- [Installing Oracle Trace File Analyzer on Linux or UNIX as Non-root User in Non-Daemon Mode](#)
If you are unable to install as `root`, then install Oracle Trace File Analyzer as the Oracle home owner.
- [Installing Oracle Trace File Analyzer on Microsoft Windows](#)
- [Installing Oracle Trace File Analyzer on Microsoft Windows in Non-Daemon Mode](#)
- [Oracle Trace File Analyzer Key Directories](#)
Based on your installation type, the `ora_home` and the `bin` directories can differ.
- [Oracle Trace File Analyzer Command Interfaces](#)
The `tfactl` tool functions as command-line interface, shell interface, and menu interface.
- [Masking Sensitive Data](#)
Masking sensitive data is an optional feature that you can configure Oracle Trace File Analyzer to mask sensitive data in log files.
- [Securing Access to Oracle Trace File Analyzer](#)
Running `tfactl` commands is restricted to authorized users.
- [Uninstalling Oracle Trace File Analyzer](#)

14.1 Supported Environments

You can use Oracle Trace File Analyzer with all supported versions of Oracle Database and Oracle Clusterware.

Oracle Trace File Analyzer works on the following operating systems:

- Linux OEL
- Linux RedHat
- Linux SuSE
- Linux Itanium
- zLinux

- Oracle Solaris SPARC
- Oracle Solaris x86-64
- AIX
- HPUX Itanium
- HPUX PA-RISC
- Microsoft Windows 64-bit

Oracle Trace File Analyzer is supported on the operating system versions supported by Oracle Database. Use a Java Runtime Edition of version 1.8.

The Oracle Clusterware install is shipped with Oracle Trace File Analyzer since versions 11.2.0.4 and 12.1.0.2. However, this install does not include many of the Oracle Database tools. Oracle releases new versions of Oracle Trace File Analyzer several times a year. These new releases include new features and bug fixes.

Ensure that you get the latest Oracle Trace File Analyzer with Oracle Database support tools bundle from My Oracle Support note 1513912.1.

Related Topics

- <https://support.oracle.com/rs?type=doc&id=1513912.1>

14.2 Installing Oracle Trace File Analyzer on Linux or UNIX as root User in Daemon Mode

To obtain the fullest capabilities of Oracle Trace File Analyzer, install it as `root`.

Oracle Trace File Analyzer maintains Access Control Lists (ACLs) to determine which users are allowed access. By default, the `GRID_HOME` owner and `ORACLE_HOME` owner have access to their respective diagnostics. No other users can perform diagnostic collections.

If Oracle Trace File Analyzer is already installed, then reinstalling performs an upgrade to the existing location. If Oracle Trace File Analyzer is not already installed, then the recommended location is `/opt/oracle.tfa`.

To install as `root`:

1. Download appropriate Oracle Trace File Analyzer zipped file, copy the downloaded file to the required machine, and then unzip.
2. Run the `installTFA` command:

```
$ ./installTFAplatform
```

The installation prompts you to do a local or cluster install.

Cluster install requires passwordless SSH user equivalency for `root` to all cluster nodes. If not already configured, then the installation optionally sets up passwordless SSH user equivalency and then removes at the end.

If you do not wish to use passwordless SSH, then you install on each host using a local install. Run the `tfactl syncnodes` command to generate and deploy relevant SSL certificates.

The Cluster Ready Services (CRS) do not manage Oracle Trace File Analyzer because Oracle Trace File Analyzer must be available if CRS go down.

The installation configures Oracle Trace File Analyzer for auto-start. The implementation of auto-start is platform-dependent. Linux uses `init`, or an `init` replacement, such as `upstart` or `systemd`. Microsoft Windows uses a Windows service.

Related Topics

- [Securing Access to Oracle Trace File Analyzer](#)
Running `tfactl` commands is restricted to authorized users.

14.3 Installing Oracle Trace File Analyzer on Linux or UNIX as Non-root User in Non-Daemon Mode

If you are unable to install as `root`, then install Oracle Trace File Analyzer as the Oracle home owner.

Oracle Trace File Analyzer has reduced capabilities in this installation mode.

You cannot complete the following tasks:

- Automate diagnostic collections
- Collect diagnostics from remote hosts
- Collect files that are not readable by the Oracle home owner, for example, `/var/log/messages`, or certain Oracle Clusterware daemon logs

To install as the Oracle home owner, use the `-extractto` option. Using the `-extractto` option tells Oracle Trace File Analyzer where to install to. Also, use the `-javahome` option to indicate which JRE to use. Use the JRE already available in the Oracle home, unless you have a later version available.

```
./installTFApatform -extractto install_dir -javahome jre_home
```

14.4 Installing Oracle Trace File Analyzer on Microsoft Windows

1. Download appropriate Oracle Trace File Analyzer zipped file, copy the downloaded file to one of the required machines, and then unzip.
2. Open a command prompt as administrator and then run the installation script by specifying a Perl home.

For example:

```
install.bat -perlhome D:\oracle\product\12.2.0\dbhome_1\perl
```

The installer prompts you to do a local or cluster install. If you select cluster install, then the installer installs Oracle Trace File Analyzer remotely on cluster nodes.

Alternatively, you can perform a local install on each host. Run the `tfactl syncnodes` command to generate and deploy relevant SSL certificates.

14.5 Installing Oracle Trace File Analyzer on Microsoft Windows in Non-Daemon Mode

If you do not want Oracle Trace File Analyzer to run automatically as a windows service, then install it in non-daemon mode. Oracle Trace File Analyzer has reduced capabilities in this installation mode.

You cannot complete the following tasks:

- Automate diagnostic collections
 - Collect diagnostics from remote hosts
 - Collect files that are not readable by the Oracle home owner
1. Download appropriate Oracle Trace File Analyzer zipped file, copy the downloaded file to one of the required machines, and then unzip.
 2. Open a command prompt as administrator and then run the installation script.

```
tfa_home\bin\tfactl.bat -setupnd
```

14.6 Oracle Trace File Analyzer Key Directories

Based on your installation type, the `ora_home` and the `bin` directories can differ.

If you have installed Oracle Trace File Analyzer with Oracle Clusterware, then `tfa_home` is within `GRID_HOME`.

Table 14-1 Key Oracle Trace File Analyzer Directories

Directory	Description
<code>tfa/bin</code>	Contains the command-line interface <code>tfactl</code> . If Oracle Clusterware is installed, then <code>tfactl</code> is also installed in the <code>GRID_HOME/bin</code> directory.
<code>tfa/repository</code>	Directory where Oracle Trace File Analyzer stores diagnostic collections.
<code>tfa/node/tfa_home/database</code>	Contains Berkeley database that stores data about the system.
<code>tfa/node/tfa_home/diag</code>	Tools for troubleshooting Oracle Trace File Analyzer.
<code>tfa/node/tfa_home/diagnostics_to_collect</code>	Place files here to include them in the next collection, then have them deleted afterwards.
<code>tfa/node/tfa_home/log</code>	Contains logs about Oracle Trace File Analyzer operation.
<code>tfa/node/tfa_home/resources</code>	Contains resource files, for example, the log masking control file.
<code>tfa/node/tfa_home/output</code>	Contains extra metadata about the environment.

14.7 Oracle Trace File Analyzer Command Interfaces

The `tfactl` tool functions as command-line interface, shell interface, and menu interface.

Table 14-2 Oracle Trace File Interfaces

Interface	Command	How to use
Command-line	<code>\$ tfactl <i>command</i></code>	Specify all command options at the command line.
Shell interface	<code>\$ tfactl</code>	Set and change the context and then run commands from within the shell.
Menu Interface	<code>\$ tfactl <i>menu</i></code>	Select the menu navigation options and then choose the command that you want to run.

14.8 Masking Sensitive Data

Masking sensitive data is an optional feature that you can configure Oracle Trace File Analyzer to mask sensitive data in log files.

Oracle Trace File Analyzer masks information such as host names or IP addresses and replaces sensitive data consistently throughout all files. Replacing consistently means that the information is still relevant and useful for the purposes of diagnosis without sharing any sensitive data.

To configure masking:

1. Create a file called `mask_strings.xml` in the directory `tfa_home/resources`.
2. Define a `mask_strings` element then within that a `mask_string` element, with *original* and *replacement* for each string you wish to replace:

For example:

```
<mask_strings>
  <mask_string>
    <original>WidgetNode1</original>
    <replacement>Node1</replacement>
  </mask_string>
  <mask_string>
    <original>192.168.5.1</original>
    <replacement>Node1-IP</replacement>
  </mask_string>
  <mask_string>
    <original>WidgetNode2</original>
    <replacement>Node2</replacement>
  </mask_string>
  <mask_string>
    <original>192.168.5.2</original>
    <replacement>Node2-IP</replacement>
  </mask_string>
</mask_strings>
```

Oracle Trace File Analyzer automatically locates the `mask_strings.xml` files, and starts replacing the sensitive data in the diagnostics it collects.

14.9 Securing Access to Oracle Trace File Analyzer

Running `tfactl` commands is restricted to authorized users.

`tfactl` provides a command-line interface and shell to do the following:

- Run diagnostics and collect all relevant log data from a time of your choosing
- Trim log files around the time, collecting only what is necessary for diagnosis
- Collect and package all trimmed diagnostics from any desired nodes in the cluster and consolidate everything in one package on a single node

Authorized non-root users can run a subset of the `tfactl` commands. All other `tfactl` commands require `root` access. Users who are not authorized cannot run any `tfactl` command.

By default, the following users are authorized to access a subset of `tfactl` commands:

- Oracle Grid Infrastructure home owner
- Oracle Database home owners

User access is applicable only if Oracle Trace File Analyzer is installed as `root` on Linux and UNIX. User access is not applicable if Oracle Trace File Analyzer is installed as non-root, or on Microsoft Windows.

To provision user access to `tfactl`:

- To list the users who have access to `tfactl`:

```
tfactl access lsusers
```

- To add a user to access `tfactl`:

```
tfactl access add -user user [-local]
```

By default, access commands apply to cluster-wide unless `-local` is used to restrict to local node.

- To remove a user from accessing `tfactl`:

```
tfactl access remove -user user [-local]
```

- To remove all users from accessing `tfactl`:

```
tfactl access removeall [-local]
```

- To reset user access to default:

```
tfactl access reset
```

- To enable user access:

```
tfactl access enable
```

- To disable user access:

```
tfactl access disable
```

Related Topics

- [tfactl access](#)

Use the `tfactl access` command to allow non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections.

14.10 Uninstalling Oracle Trace File Analyzer

1. To uninstall Oracle Trace File Analyzer, run the `uninstall` command as `root`, or install user.

```
$ tfactl uninstall
```

15

Automatic Diagnostic Collections

Oracle Trace File Analyzer monitors your logs for significant problems, such as internal errors like `ORA-00600`, or node evictions.

- [Collecting Diagnostics Automatically](#)
This section explains automatic diagnostic collection concepts.
- [Configuring Email Notification Details](#)
Configure Oracle Trace File Analyzer to send an email to the registered email address after an automatic collection completes.

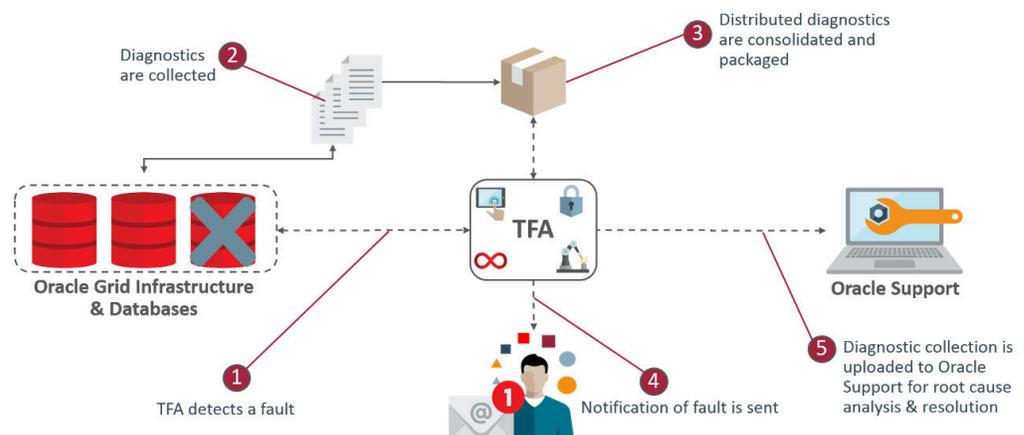
15.1 Collecting Diagnostics Automatically

This section explains automatic diagnostic collection concepts.

If Oracle Trace File Analyzer detects any problems, then it carries out the following actions:

- Runs necessary diagnostics and collects all relevant log data at the time of a problem
- Trims log files around the time of the problem so that Oracle Trace File Analyzer collects only what is necessary for diagnosis
- Collects and packages all trimmed diagnostics from all nodes in the cluster, consolidating everything on a single node
- Stores diagnostic collections in the Oracle Trace File Analyzer repository
- Sends you email notification of the problem and details of diagnostic collection that is ready for upload to Oracle Support

Figure 15-1 Automatic Diagnostic Collections



Oracle Trace File Analyzer uses a flood control mechanism. Repeated errors do not flood the system with automatic collections.

Identifying an event triggers the start point for a collection and five minutes later Oracle Trace File Analyzer starts collecting diagnostic data. Starting five minutes later is to capture any other relevant events together. If events are still occurring after five minutes, then diagnostic collection continues to wait. Oracle Trace File Analyzer waits for 30 seconds with no events occurring, up to a further five minutes.

If events are still occurring after 10 minutes, then a diagnostic collection happens. A new collection point starts.

After the collection is complete, Oracle Trace File Analyzer sends email notification that includes the location of the collection, to the relevant recipients.

If your environment can make a connection to **oracle.com**, you can then use Oracle Trace File Analyzer to upload the collection to a Service Request.

```
$ tfactl set autodiagcollect=ON|OFF
```

Automatic collections are **ON** by default.

Table 15-1 Log Entries that Trigger Automatic collection

String Pattern	Log Monitored
ORA-297(01 02 03 08 09 10 40)	Alert Log - Oracle Database
ORA-00600	Alert Log - Oracle ASM
ORA-07445	Alert Log - Oracle ASM Proxy
ORA-04(69 ([7-8][0-9] 9([0-3] [5-8])))	Alert Log - Oracle ASM IO Server
ORA-32701	
ORA-00494	
System State dumped	
CRS-016(07 10 11 12)	Alert Log - CRS

Related Topics

- [Uploading Collections to Oracle Support](#)
To enable collection uploads, configure Oracle Trace File Analyzer with your My Oracle Support user name and password.

15.2 Configuring Email Notification Details

Configure Oracle Trace File Analyzer to send an email to the registered email address after an automatic collection completes.

To send emails, configure the system on which Oracle Trace Analyzer is running. You must configure notification with a user email address to enable it to work.

To configure email notification details:

1. To set the notification email to use for a specific `ORACLE_HOME`, include the operating system owner in the command:

```
tfactl set notificationAddress=os_user:email
```

For example:

```
tfactl set notificationAddress=oracle:some.body@example.com
```

2. To set the notification email to use for any `ORACLE_HOME`:

```
tfactl set notificationAddress=email
```

For example:

```
tfactl set notificationAddress=another.body@example.com
```

3. Do the following after receiving the notification email:
 - a. To find the root cause, inspect the referenced collection details.
 - b. If you can fix the issue, then resolve the underlying cause of the problem.
 - c. If you do not know the root cause of the problem, then log an SR with Oracle Support, and upload the collection details.

16

On-demand Analysis and Diagnostic Collection

Run Oracle Trace File Analyzer on demand using `tfactl` command-line tool.

- [Collecting Diagnostics and Analyzing Logs On-Demand](#)
The `tfactl` command can use a combination of different database command tools when it performs analysis.
- [Viewing System and Cluster Summary](#)
The summary command gives you a real-time report of system and cluster status.
- [Investigating Logs for Errors](#)
Use Oracle Trace File Analyzer to analyze all your logs across your cluster to identify recent errors.
- [Analyzing Logs Using the Included Tools](#)
Oracle Database support tools bundle is available only when you download Oracle Trace File Analyzer from My Oracle Support note 1513912.1.
- [Collecting Diagnostic Data and Using One Command Service Request Data Collections](#)
- [Uploading Collections to Oracle Support](#)
To enable collection uploads, configure Oracle Trace File Analyzer with your My Oracle Support user name and password.
- [Changing Oracle Clusterware Trace Levels](#)
Enabling trace levels enables you to collect enough diagnostics to diagnose the cause of the problem.

16.1 Collecting Diagnostics and Analyzing Logs On-Demand

The `tfactl` command can use a combination of different database command tools when it performs analysis.

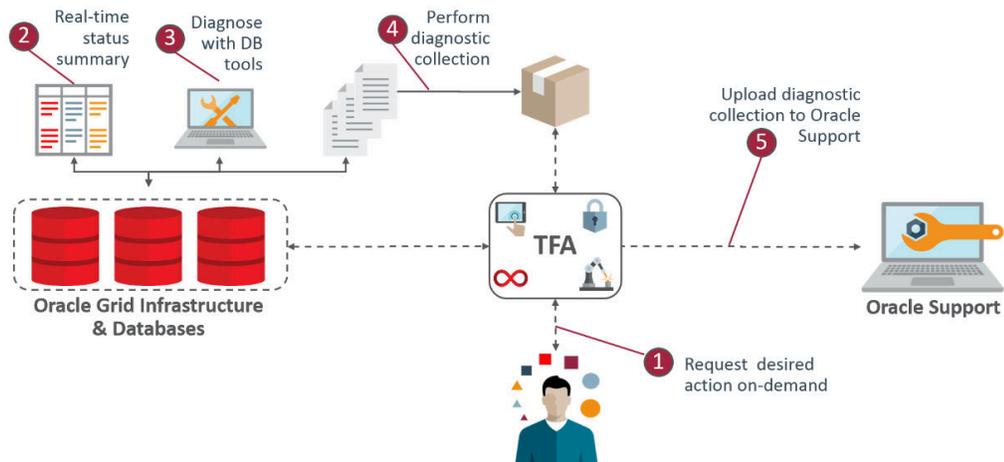
The `tfactl` command enables you to access all tools using common syntax. Using common syntax hides the complexity of the syntax differences between the tools.

Use the Oracle Trace File Analyzer tools to perform analysis and resolve problems. If you need more help, then use the `tfactl` command to collect diagnostics for Oracle Support.

Oracle Trace File Analyzer does the following:

- Collects all relevant log data from a time of your choosing.
- Trims log files around the time, collecting only what is necessary for diagnosis.
- Packages all diagnostics on the node where `tfactl` was run from.

Figure 16-1 On-Demand Collections



16.2 Viewing System and Cluster Summary

The summary command gives you a real-time report of system and cluster status.

Syntax

```
tfactl summary [options]
```

For more help use:

```
tfactl summary -help
```

16.3 Investigating Logs for Errors

Use Oracle Trace File Analyzer to analyze all your logs across your cluster to identify recent errors.

1. To find all errors in the last one day:

```
$ tfactl analyze -last 1d
```

2. To find all errors over a specified duration:

```
$ tfactl analyze -last 18h
```

3. To find all occurrences of a specific error on any node, for example, to report ORA-00600 errors:

```
$ tfactl analyze -search "ora-00600" -last 8h
```

Example 16-1 Analyzing logs

```
tfactl analyze -last 14d
```

```
Jun/02/2016 11:44:39 to Jun/16/2016 11:44:39 tfactl> analyze -last 14d
INFO: analyzing all (Alert and Unix System Logs) logs for the last 20160 minutes...
Please wait...
INFO: analyzing host: myserver69
```

```
Report title: Analysis of Alert, System Logs
```

```

Report date range: last ~14 day(s)
Report (default) time zone: EST - Eastern Standard Time
Analysis started at: 16-Jun-2016 02:45:02 PM EDT
Elapsed analysis time: 0 second(s).
Configuration file:
/u01/app/tfa/myserver69/tfa_home/ext/tnt/conf/tnt.prop
Configuration group: all
Total message count:          957, from 02-May-2016
09:04:07 PM EDT to 16-Jun-2016 12:45:41 PM EDT
Messages matching last ~14 day(s): 225, from 03-Jun-2016
02:17:32 PM EDT to 16-Jun-2016 12:45:41 PM EDT
last ~14 day(s) error count:          2, from 09-Jun-2016
09:56:47 AM EDT to 09-Jun-2016 09:56:58 AM EDT last ~14 day(s) ignored error count: 0
last ~14 day(s) unique error count: 2

```

```

Message types for last ~14 day(s)
Occurrences percent  server name          type
-----
223  99.1%  myserver69          generic
2    0.9%  myserver69          ERROR
-----
225  100.0%

```

```

Unique error messages for last ~14 day(s)
Occurrences percent  server name          error
-----
1    50.0%  myserver69          Errors in file
/u01/app/racusr/diag/rdbms/rdb11204/RDB112041/trace/RDB112041_ora_25401.trc
(incident=6398):

```

```

ORA-07445: exception
encountered: core dump [] [] [] [] [] []
Incident details in:
/u01/app/racusr/diag/rdbms/rdb11204/RDB112041/incident/incdir_6398/
RDB112041_ora_25401_i6398.trc

Use ADRCI or Support Workbench to
package the incident.

See Note 411.1 at My Oracle Support
for error and packaging details.

```

```

1    50.0%  myserver69          Errors in file
/u01/app/racusr/diag/rdbms/rdb11204/RDB112041/trace/RDB112041_ora_25351.trc
(incident=6394):
ORA-00700: soft internal error,
arguments: [kgerev1], [600], [600], [700], [], [], [], [], [], [], [], []
Incident details in:
/u01/app/racusr/diag/rdbms/rdb11204/RDB112041/incident/incdir_6394/
RDB112041_ora_25351_i6394.trc

```

```

Errors in file /u01/app/racusr/diag/
rdbms/rdb11204/RDB112041/trace/RDB112041_ora_25351.trc
(incident=6395):
ORA-00600: internal error code,
arguments: [], [], [], [], [], [], [], [], [], [], [], []
Incident details in:
/u01/app/racusr/diag/rdbms/rdb11204/RDB112041/incident/incdir_6395/
RDB112041_ora_25351_i6395.trc

```

```

Dumping diagnostic data in
directory=[cdmp_20160609095648], requested by (instance=1, osid=25351),
summary=[incident=6394].

```

package the incident. Use ADRCI or Support Workbench to
for error and packaging details. See Note 411.1 at My Oracle Support

```
-----
      2 100.0%
See Change Which Directories Get Collected for more details.
```

Related Topics

- [tfactl summary](#)
Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.
- [tfactl analyze](#)
Use the `tfactl analyze` command to obtain analysis of your system by parsing the database, Oracle ASM, and Oracle Grid Infrastructure alert logs, system message logs, OSWatcher Top, and OSWatcher Slabinfo files.

16.4 Analyzing Logs Using the Included Tools

Oracle Database support tools bundle is available only when you download Oracle Trace File Analyzer from My Oracle Support note 1513912.1.

Oracle Trace File Analyzer with Oracle Database support tools bundle includes the following tools:

Table 16-1 Tools included in Linux and UNIX

Tool	Description
<code>orachk</code> or <code>exachk</code>	Provides health checks for the Oracle stack. Oracle Trace File Analyzer installs either Oracle EXAchk for engineered systems or Oracle ORAchk for all non-engineered systems. For more information, see My Oracle Support notes 1070954.1 and 1268927.2.
<code>oswatcher</code>	Collects and archives operating system metrics. These metrics are useful for instance or node evictions and performance Issues. For more information, see My Oracle Support note 301137.1.
<code>proctwatcher</code>	Automates and captures database performance diagnostics and session level hang information. For more information, see My Oracle Support note 459694.1.
<code>oratop</code>	Provides near real-time database monitoring. For more information, see My Oracle Support note 1500864.1.
<code>alertsummary</code>	Provides summary of events for one or more database or ASM alert files from all nodes.
<code>ls</code>	Lists all files Oracle Trace File Analyzer knows about for a given file name pattern across all nodes.
<code>pstack</code>	Generates the process stack for the specified processes across all nodes.
<code>grep</code>	Searches for a given string in the alert or trace files with a specified database.

Table 16-1 (Cont.) Tools included in Linux and UNIX

Tool	Description
summary	Provides high-level summary of the configuration.
vi	Opens alert or trace files for viewing a given database and file name pattern in the <code>vi</code> editor.
tail	Runs a tail on an alert or trace files for a given database and file name pattern.
param	Shows all database and operating system parameters that match a specified pattern.
dbglevel	Sets and unsets multiple CRS trace levels with one command.
history	Shows the shell history for the <code>tfactl</code> shell.
changes	Reports changes in the system setup over a given time period. The report includes database parameters, operating system parameters, and the patches applied.
calog	Reports major events from the cluster event log.
events	Reports warnings and errors seen in the logs.
managelogs	Shows disk space usage and purges ADR log and trace files.
ps	Finds processes.
trriage	Summarizes <code>oswatcher</code> or <code>exawatcher</code> data.

Table 16-2 Tools included in Microsoft Windows

Tool	Description
calog	Reports major events from the cluster event log.
changes	Reports changes in the system setup over a given time period. The report includes database parameters, operating system parameters, and patches applied.
dir	Lists all files Oracle Trace File Analyzer knows about for a given file name pattern across all nodes.
events	Reports warnings and errors seen in the logs.
findstr	Searches for a given string in the alert or trace files with a specified database.
history	Shows the shell history for the <code>tfactl</code> shell.
managelogs	Shows disk space usage and purges ADR log and trace files.
notepad	Opens alert or trace files for viewing a given database and file name pattern in the <code>notepad</code> editor.
param	Shows all database and operating system parameters that match a specified pattern.
summary	Provides high-level summary of the configuration.
tasklist	Finds processes.

To verify which tools you have installed:

```
$ tfactl toolstatus
```

You can run each tool using `tfactl` either in command line or shell mode.

To run a tool from the command line:

```
$ tfactl run tool
```

The following example shows how to use `tfactl` in shell mode. Running the command starts `tfactl`, connects to the database `MyDB`, and then runs `oratop`:

```
$ tfactl
tfactl > database MyDB
MyDB tfactl > oratop
```

Related Topics

- <https://support.oracle.com/rs?type=doc&id=1513912.1>
- <https://support.oracle.com/rs?type=doc&id=1070954.1>
- <https://support.oracle.com/rs?type=doc&id=1268927.2>
- <https://support.oracle.com/rs?type=doc&id=301137.1>
- <https://support.oracle.com/rs?type=doc&id=459694.1>
- <https://support.oracle.com/rs?type=doc&id=1500864.1>
- <https://support.oracle.com/rs?type=doc&id=215187.1>

16.5 Collecting Diagnostic Data and Using One Command Service Request Data Collections

To perform an on-demand diagnostic collection:

```
$ tfactl diagcollect
```

Running the command trims and collects all important log files updated in the past 12 hours across the whole cluster. Oracle Trace File Analyzer stores collections in the repository directory. You can change the `diagcollect` timeframe with the `-last n h|d` option.

Oracle Support often asks you to run a Service Request Data Collection (SRDC). The SRDC depends on the type of problem you experienced. It is a series of many data gathering instructions aimed at diagnosing your problem. Collecting the SRDC manually can be difficult, with many different steps required.

Oracle Trace File Analyzer can run SRDC collections with a single command:

```
$ tfactl diagcollect -srdc srdc_type -sr sr_number
```

To run SRDCs, use one of the Oracle privileged user accounts:

- `ORACLE_HOME owner`
- `GRID_HOME owner`

Table 16-3 One Command Service Request Data Collections

Type of Problem	Available SRDCs	Collection Scope	
ORA Errors	ORA-000 20	ORA-0403 1	Local-only
	ORA-000 60	ORA-0744 5	
	ORA-006 00	ORA-2730 0	
	ORA-007 00	ORA-2730 1	
	ORA-015 55	ORA-2730 2	
	ORA-016 28	ORA-3003 6	
	ORA-040 30		
	Other internal database errors	internalerror	
Database performance problems	dbperf	Cluster-wide	
Database patching problems	dbpatchinstall	Local-only	
	dbpatchconflict		
Database install / upgrade problems	dbinstall	Local-only	
	dbupgrade		
	dbpreupgrade		
Database storage problems	asm	Local-only	
Excessive SYSAUX space is used by the Automatic Workload Repository (AWR)	dbawrspace	Local-only	
Database startup / shutdown problems	dbshutdown		
	dbstartup		
Enterprise Manager tablespace usage metric problems	emtbmetrics	Local-only (on Enterprise Manager Agent target)	
Enterprise Manager general metrics page or threshold problems	emmetricalert	Local-only (on Enterprise Manager Agent target and repository database)	
Enterprise Manager debug log collection	emdebugon	Local-only (on Enterprise Manager Agent target and Oracle Management Service)	
Run emdebugon, reproduce the problem then run emdebugoff, which disables debug again and collects debug logs	emdebugoff		
Data Guard problems	dbdataguard	Local-only	

What the SRDCs collect varies for each type, for example:

Table 16-4 SRDC collections

Command	What gets collected
\$ tfactl diagcollect -srdc ORA-04031	<ul style="list-style-type: none"> • IPS package • Patch listing • AWR report • Memory information • RDA HCVE output
\$ tfactl diagcollect -srdc dbperf	<ul style="list-style-type: none"> • ADDM report • AWR for good period and problem period • AWR Compare Period report • ASH report for good and problem period • OSWatcher • IPS package (if there are any errors during problem period) • Oracle ORAchk (performance-related checks)

Oracle Trace File Analyzer prompts you to enter the information required based on the SRDC type.

For example, when you run ORA-4031 SRDC:

```
$ tfactl diagcollect -srdc ORA-04031
```

Oracle Trace File Analyzer prompts to enter event date/time and database name.

1. Oracle Trace File Analyzer scans the system to identify recent events in the system (up to 10).
2. Once the relevant event is chosen, Oracle Trace File Analyzer then proceeds with diagnostic collection.
3. Oracle Trace File Analyzer identifies all the required files.
4. Oracle Trace File Analyzer trims all the files where applicable.
5. Oracle Trace File Analyzer packages all data in a zip file ready to provide to support.

You can also run an SRDC collection in non-interactive silent mode. Provide all the required parameters up front as follows:

```
$ tfactl diagcollect -srdc srdc_type -database db -from "date time" -to "date time"
```

Example 16-2 Diagnostic Collection

```
$ tfactl diagcollect
```

```
Collecting data for the last 12 hours for all components...
Collecting data for all nodes
```

```
Collection Id : 20160616115923myserver69
```

```
Detailed Logging at :
```

```
/u01/app/tfa/repository/collection_Thu_Jun_16_11_59_23_PDT_2016_node_all/
diagcollect_20160616115923_myserver69.log
2016/06/16 11:59:27 PDT : Collection Name :
tfa_Thu_Jun_16_11_59_23_PDT_2016.zip
```

```

2016/06/16 11:59:28 PDT : Collecting diagnostics from hosts :
[myserver70, myserver71, myserver69]
2016/06/16 11:59:28 PDT : Scanning of files for Collection in progress...
2016/06/16 11:59:28 PDT : Collecting additional diagnostic information...
2016/06/16 11:59:33 PDT : Getting list of files satisfying time range
[06/15/2016 23:59:27 PDT, 06/16/2016 11:59:33 PDT]
2016/06/16 11:59:37 PDT : Collecting ADR incident files...
2016/06/16 12:00:32 PDT : Completed collection of additional diagnostic
information...
2016/06/16 12:00:39 PDT : Completed Local Collection
2016/06/16 12:00:40 PDT : Remote Collection in Progress...

```

```

-----
|           Collection Summary           |
+-----+-----+-----+-----+
| Host       | Status    | Size  | Time  |
+-----+-----+-----+-----+
| myserver71 | Completed | 15MB  | 64s   |
| myserver70 | Completed | 14MB  | 67s   |
| myserver69 | Completed | 14MB  | 71s   |
+-----+-----+-----+-----+

```

```

Logs are being collected to:
/u01/app/tfa/repository/collection_Thu_Jun_16_11_59_23_PDT_2016_node_all
/u01/app/tfa/repository/collection_Thu_Jun_16_11_59_23_PDT_2016_node_all/
myserver71.tfa_Thu_Jun_16_11_59_23_PDT_2016.zip
/u01/app/tfa/repository/collection_Thu_Jun_16_11_59_23_PDT_2016_node_all/
myserver69.tfa_Thu_Jun_16_11_59_23_PDT_2016.zip
/u01/app/tfa/repository/collection_Thu_Jun_16_11_59_23_PDT_2016_node_all/
myserver70.tfa_Thu_Jun_16_11_59_23_PDT_2016.zip

```

Example 16-3 One command SRDC

```

$ tfactl diagcollect -srdc ora600
Enter value for EVENT_TIME [YYYY-MM-DD HH24:MI:SS,<RETURN>=ALL] :
Enter value for DATABASE_NAME [<RETURN>=ALL] :

1. Jun/09/2016 09:56:47 : [rdb11204] ORA-00600: internal error code,
arguments: [], [], [], [], [], [], [], [], [], [], [], [] 2. May/19/2016 14:19:30 :
[rdb11204] ORA-00600: internal error code,
arguments: [], [], [], [], [], [], [], [], [], [], [], [] 3. May/13/2016 10:14:30 :
[rdb11204] ORA-00600: internal error code,
arguments: [], [], [], [], [], [], [], [], [], [], [], [] 4. May/13/2016 10:14:09 :
[rdb11204] ORA-00600: internal error code,
arguments: [], [], [], [], [], [], [], [], [], [], [], []

Please choose the event : 1-4 [1] 1
Selected value is : 1 ( Jun/09/2016 09:56:47 ) Collecting data for local node(s)
Scanning files
from Jun/09/2016 03:56:47 to Jun/09/2016 15:56:47

Collection Id : 20160616115820myserver69

Detailed Logging at :
/u01/app/tfa/repository/
srdc_ora600_collection_Thu_Jun_16_11_58_20_PDT_2016_node_local/
diagcollect_20160616115820_myserver69.log
2016/06/16 11:58:23 PDT : Collection Name :
tfa_srdc_ora600_Thu_Jun_16_11_58_20_PDT_2016.zip
2016/06/16 11:58:23 PDT : Scanning of files for Collection in progress...
2016/06/16 11:58:23 PDT : Collecting additional diagnostic information...
2016/06/16 11:58:28 PDT : Getting list of files satisfying time range

```

```
[06/09/2016 03:56:47 PDT, 06/09/2016 15:56:47 PDT]
2016/06/16 11:58:30 PDT : Collecting ADR incident files...
2016/06/16 11:59:02 PDT : Completed collection of additional diagnostic
information...
2016/06/16 11:59:06 PDT : Completed Local Collection
-----
|           Collection Summary           |
+-----+-----+-----+-----+
| Host       | Status   | Size    | Time    |
+-----+-----+-----+-----+
| myserver69 | Completed | 7.9MB   | 43s    |
+-----+-----+-----+-----+
```

16.6 Uploading Collections to Oracle Support

To enable collection uploads, configure Oracle Trace File Analyzer with your My Oracle Support user name and password.

For example:

```
tfactl setupmos
```

Oracle Trace File Analyzer stores your login details securely within an encrypted wallet. You can store only a single user login details.

1. Run a diagnostic collection using the `-sr sr_number` option.

```
tfactl diagcollect diagcollect options -sr sr_number
```

At the end of collection, Oracle Trace File Analyzer automatically uploads all collections to your Service Request.

Oracle Trace File Analyzer can also upload any other file to your Service Request.

You can upload using the wallet, which was setup previously by `root` using `tfactl setupmos`.

```
tfactl upload -sr sr_number -wallet space-separated list of files to upload
```

You can also upload without the wallet. When uploading without the wallet `tfactl` prompts for the password.

```
tfactl upload -sr sr_number -user user_id space-separated list of files to upload
```

16.7 Changing Oracle Clusterware Trace Levels

Enabling trace levels enables you to collect enough diagnostics to diagnose the cause of the problem.

Oracle Support asks you to enable certain trace levels when reproducing a problem.

Oracle Trace File Analyzer makes it easy to enable and then disable the correct trace levels. Use the `dbglevel` option to set the trace level.

You can find the required trace level settings grouped by problem trace profiles.

To set trace levels:

1. To set a trace profile:

```
tfactl dbglevel -set profile
```

2. To list all available profiles:

```
tfactl dbglevel -help
```

- **tfactl dbglevel**

Use the `tfactl dbglevel` command to set Oracle Clusterware trace levels.

16.7.1 tfactl dbglevel

Use the `tfactl dbglevel` command to set Oracle Clusterware trace levels.

Syntax

```
tfactl dbglevel [ {-set|-unset} profile_name
-dependency [dep1, dep2,...|all]
-dependency_type [type1, type2, type3,...|all]
| {-view|-drop} profile_name | -lsprofiles | -lsmodules | -lscomponents
[module_name]
| -lsres | -create profile_name [ -desc description
| [-includeunset] [-includetrace] | -debugstate ] | -modify profile_name
[-includeunset] [-includetrace] | -getstate [ -module module_name ]
| -active [profile_name] | -describe [profile_name] ] ]
```

Parameters

Table 16-5 tfactl dbglevel Command Parameters

Parameter	Description
profile_name	Specify the name of the profile.
active	Displays the list of active profiles.
set	Sets the trace or log levels for the profile specified.
unset	Unsets the trace or log levels for the profile specified.
view	Displays the trace or log entries for the profile specified.
create	Creates a profile.
drop	Drops the profile specified.
modify	Modifies the profile specified.
describe	Describes the profiles specified.
lsprofiles	Lists all the available profiles.
lsmodules	Lists all the discovered CRS modules.
lscomponents	Lists all the components associated with the CRS module.
lsres	Lists all the discovered CRS resources.
getstate	Displays the current trace or log levels for the CRS components or resources.
module	Specify the CRS module.
dependency	Specify the dependencies to consider, start, or stop dependencies, or both.
dependency_type	Specify the type of dependencies to be consider.

Table 16-5 (Cont.) tfactl dbglevel Command Parameters

Parameter	Description
debugstate	Generates a System State Dump for all the available levels.
includeunset	Adds or modifies an unset value for the CRS components or resources.
includetrace	Adds or modifies a trace value for the CRS components.

 **WARNING:**

Set the profiles only at the direction of Oracle Support.

17

Maintaining Oracle Trace File Analyzer to the Latest Version

Oracle releases a new version of Oracle Trace File Analyzer approximately every three months.

Applying standard Release Update Revisions (RURs) automatically update Oracle Trace File Analyzer. However, the Release Update Revisions (RURs) do not contain the rest of the Oracle Database support tools bundle updates. Download the latest version of Oracle Trace File Analyzer with Oracle Database support tools bundle from My Oracle Support note 1513912.1.

Upgrading is similar to first-time install. As `root`, use the `installTFPlatform` script. If Oracle Trace File Analyzer is already installed, then the installer updates the existing installation. When already installed, a cluster upgrade does not need SSH. The cluster upgrade uses the existing daemon secure socket communication between hosts.

```
$ ./installTFPlatform
```

If you are not able to install as `root`, then install Oracle Trace File Analyzer as Oracle home owner. Use the `-extractto` and `-javahome` options:

```
$ ./installTFPlatform -extractto dir -javahome jre_home
```

Related Topics

- [Installing Oracle Trace File Analyzer on Microsoft Windows](#)
- <https://support.oracle.com/rs?type=doc&id=1513912.1>

18

Performing Custom Collections

Use the custom collection options to change the diagnostic collections from the default.

- [Adjusting the Diagnostic Data Collection Period](#)
Oracle Trace File Analyzer trims and collects any important logs updated in the past 12 hours.
- [Collecting from Specific Nodes](#)
- [Collecting from Specific Components](#)
- [Collecting from Specific Directories](#)
- [Changing the Collection Name](#)
- [Preventing Copying Zip Files and Trimming Files](#)
- [Performing Silent Collection](#)
- [Preventing Collecting Core Files](#)
- [Collecting Incident Packaging Service \(IPS\) Packages](#)
Incident Packaging Service packages details of problems stored by Oracle Database in ADR for later diagnosis.

18.1 Adjusting the Diagnostic Data Collection Period

Oracle Trace File Analyzer trims and collects any important logs updated in the past 12 hours.

If you know that you only want logs for a smaller window, then you can cut this collection period. Cutting the collection period helps you make collections as small and quick as possible.

There are four different ways you can specify the period for collection:

Table 18-1 Ways to Specify the Collection Period

Command	Description
<code>tfactl diagcollect -last n h d</code>	Collects since the previous <i>n</i> hours or days.
<code>tfactl diagcollect -from "yyyy-mm-dd"</code>	Collects from the date and optionally time specified. Valid date and time formats: "Mon/dd/yyyy hh:mm:ss" "yyyy-mm-dd hh:mm:ss" "yyyy-mm-ddThh:mm:ss" "yyyy-mm-dd"

Table 18-1 (Cont.) Ways to Specify the Collection Period

Command	Description
<code>tfactl diagcollect -from "yyyy-mm-dd" -to "yyyy-mm-dd"</code>	Collects between the date and optionally time specified. Valid date and time formats: "Mon/dd/yyyy hh:mm:ss" "yyyy-mm-dd hh:mm:ss" "yyyy-mm-ddThh:mm:ss" "yyyy-mm-dd"
<code>tfactl diagcollect -for "yyyy-mm-dd"</code>	Collects for the specified date. Valid date formats: "Mon/dd/yyyy" "yyyy-mm-dd"

18.2 Collecting from Specific Nodes

To collect from specific nodes:

1. To collect from specific nodes:

```
tfactl diagcollect -node list of nodes
```

For example:

```
$ tfactl diagcollect -last 1d -node myserver65
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.3 Collecting from Specific Components

To collect from specific components:

1. To collect from specific components:

```
tfactl diagcollect component
```

For example:

To trim and collect all files from the databases `hrdb` and `fdb` in the last 1 day:

```
$ tfactl -diagcollect -database hrdb,fdb -last 1d
```

To trim and collect all CRS files, operating system logs, and CHMOS/OSW data from `node1` and `node2` updated in the last 6 hours:

```
$ tfactl diagcollect -crs -os -node node1,node2 -last 6h
```

To trim and collect all Oracle ASM logs from `node1` updated between from and to time:

```
$ tfactl diagcollect -asm -node node1 -from "2016-08-15" -to "2016-08-17"
```

Following are the available component options.

Table 18-2 Component Options

Component Option	Description
-database <i>database_names</i>	Collects database logs from databases specified in a comma-separated list.
-asm	Collects Oracle ASM logs.
-crsclient	Collects Client Logs that are under <code>GIBASE/diag/clients</code> .
-dbclient	Collects Client Logs that are under <code>DB ORABASE/diag/clients</code> .
-dbwlm	Collects DBWLM logs.
-tns	Collects TNS logs.
-rhp	Collects RHP logs.
-procinfo	Collects <code>Gathers stack</code> and <code>fd</code> from <code>/proc</code> for all processes.
-afd	Collects AFD logs.
-crs	Collects CRS logs.
-wls	Collects WLS logs.
-emagent	Collects EMAGENT logs.
-oms	Collects OMS logs.
-ocm	Collects OCM logs.
-emplugins	Collects EMPLUGINS logs.
-em	Collects EM logs.
-acfs	Collects ACFS logs and data.
-install	Collects Oracle Installation related files.
-cfgtools	Collects CFGTOOLS logs.
-os	Collects operating system files such as <code>/var/log/messages</code> .
-ashhtml	Collects Generate ASH HTML Report.
-ashtext	Collects Generate ASH TEXT Report.
-awrhtml	Collects AWRHTML logs.

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.4 Collecting from Specific Directories

Oracle Trace File Analyzer discovers all Oracle diagnostics and collects relevant files based on the type and last time updated.

If you want to collect other files, then you can specify extra directories. Oracle Trace File Analyzer collects only the files updated in the relevant time range (12 hours by default).

You can configure collection of all files irrespective of the time last updated. Configure on a directory by directory basis using the `-collectall` option.

To collect from specific directories:

1. To include all files updated in the last 12 hours:

```
tfactl diagcollect -collectedir dir1,dir2,...dirn
```

For example:

To trim and collect all CRS files updated in the last 12 hours as well as all files from `/tmp_dir1` and `/tmp_dir2` at the initiating node:

```
$ tfactl diagcollect -crs -collectedir /tmp_dir1,/tmpdir_2
```

2. To configure Oracle Trace File Analyzer to collect all files from a directory, first configure it with the `-collectall` option:

```
$ tfactl add dir -collectall
```

or

```
tfactl modify dir -collectall
```

Start a diagnostic collection using the `-collectalldirs` option:

```
$ tfactl diagcollect -collectalldirs
```

Note:

If the `-collectalldirs` option is not used normal, then the file type, name, and time range restrictions are applied.

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.5 Changing the Collection Name

Oracle Trace File Analyzer zips collections and puts the zip files in the repository directory, using the following naming format:

```
repository/collection_date_time/node_all/node.tfa_date_time.zip
```

You must only change the name of the zipped files using the following options. Manually changing the file name prevents you from using collections with various Oracle Support self-service tools.

To change the collection name:

1. To use your own naming to organize collections:

```
-tag tagname
```

The files are collected into *tagname* directory inside the repository.

For example:

```
$ tfactl diagcollect -last 1h -tag MyTagName
Collecting data for all nodes
....
....
```

```
Logs are being collected to: /scratch/app/crsusr/tfa/repository/MyTagName/
/scratch/app/crsusr/tfa/repository/MyTagName/
rws1290666.tfa_Mon_Aug_22_05_26_17_PDT_2016.zip
/scratch/app/crsusr/tfa/repository/MyTagName/
rws1290665.tfa_Mon_Aug_22_05_26_17_PDT_2016.zip
```

2. To rename the zip file:

```
-z zip name
```

For example:

```
$ tfactl diagcollect -last 1h -z MyCollectionName.zip
Collecting data for all nodes
....
....
```

```
Logs are being collected to: /scratch/app/crsusr/tfa/repository/
collection_Mon_Aug_22_05_13_41_PDT_2016_node_all
/scratch/app/crsusr/tfa/repository/
collection_Mon_Aug_22_05_13_41_PDT_2016_node_all/
myserver65.tfa_MyCollectionName.zip
/scratch/app/crsusr/tfa/repository/
collection_Mon_Aug_22_05_13_41_PDT_2016_node_all/
myserver66.tfa_MyCollectionName.zip
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.6 Preventing Copying Zip Files and Trimming Files

By default, Oracle Trace File Analyzer Collector:

- Copies back all zip files from remote nodes to the initiating node
- Trims files around the relevant time

To prevent copying zip files and trimming files:

1. To prevent copying the zip file back to the initiating node:

```
-nocopy
```

For example:

```
$ tfactl diagcollect -last 1d -nocopy
```

2. To avoid trimming files:

```
-notrim
```

For example:

```
$ tfactl diagcollect -last 1d -notrim
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.7 Performing Silent Collection

1. To initiate a silent collection:

```
-silent
```

The `diagcollect` command is submitted as a background process.

For example:

```
$ tfactl diagcollect -last 1d -silent
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.8 Preventing Collecting Core Files

1. To prevent core files being included:

```
-nocores
```

For example:

```
$ tfactl diagcollect -last 1d -nocores
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

18.9 Collecting Incident Packaging Service (IPS) Packages

Incident Packaging Service packages details of problems stored by Oracle Database in ADR for later diagnosis.

Oracle Trace File Analyzer runs IPS to query and collect these packages.

Syntax

```
tfactl ips option
```

Table 18-3 `tfactl ips` Command Parameters

Command	Description
<code>tfactl ips</code>	Runs the IPS.

Table 18-3 (Cont.) tfactl ips Command Parameters

Command	Description
<code>tfactl ips show incidents</code>	Shows all IPS incidents.
<code>tfactl ips show problems</code>	Shows all IPS problems.
<code>tfactl ips show package</code>	Shows all IPS Packages.
<code>tfactl diagcollect -ips -h</code>	Shows all available <code>diagcollect</code> IPS options.
<code>tfactl diagcollect -ips</code>	Performs an IPS collection following prompts. You can use all the standard <code>diagcollect</code> options to limit the scope of IPS collection.
<code>tfactl diagcollect -ips - adrbasepath <i>adr_base</i> - adrhomepath <i>adr_home</i></code>	Performs an IPS collection in silent mode.
<code>tfactl diagcollect -ips - incident <i>incident_id</i></code>	Collects ADR details about a specific incident id.
<code>tfactl diagcollect -ips - problem <i>problem_id</i></code>	Collect ADR details about a specific problem id.

You can change the contents of the IPS package. Use the following options:

1. Start the collection.
2. Suspend the collection using the `-manageips` option.

For example:

```
$ tfactl diagcollect -ips -incident incident_id -manageips -node local
```

3. Find the suspended collection using the `print suspendedips` option.

For example:

```
$ tfactl print suspendedips
```

4. Manipulate the package.
5. Resume the collection using the `-resumeips` option.

For example:

```
$ tfactl diagcollect -resumeips collection_id
```

Example 18-1 Show Incidents

```
$ tfactl ips show incidents

ADR Home = /scratch/app/crsusr/diag/clients/user_crsusr/host_622665046_106:
*****
0 rows fetched

ADR Home = /scratch/app/crsusr/diag/afdbot/user_root/host_622665046_106:
*****
0 rows fetched

ADR Home = /scratch/app/crsusr/diag/rdbms/_mgmtdb/-MGMTDB:
*****
```

```

INCIDENT_ID PROBLEM_KEY CREATE_TIME
-----
-----

12913 ORA 700 [kskvstatact: excessive swapping observed] 2016-06-30 14:05:48.491000
-07:00

12914 ORA 700 [kskvstatact: excessive swapping observed] 2016-06-30 15:06:16.545000
-07:00

13161 ORA 445 2016-06-30 15:10:53.756000 -07:00

ADR Home = /scratch/app/crsusr/diag/asm/+asm/+ASM1:

```

```

INCIDENT_ID PROBLEM_KEY CREATE_TIME
-----
-----

1177 ORA 445 2016-06-30 15:10:12.930000 -07:00

ADR Home = /scratch/app/crsusr/diag/asm/user_root/host_622665046_106:

```

Example 18-2 Show Problems

```

$ tfactl ips show problems

ADR Home = /scratch/app/crsusr/diag/afdbboot/user_root/host_622665046_106:
*****

0 rows fetched

ADR Home = /scratch/app/crsusr/diag/rdbms/_mgmtdb/-MGMTDB:

```

```

PROBLEM_ID PROBLEM_KEY LAST_INCIDENT LASTINC_TIME
-----
-----

1 ORA 700 [kskvstatact: excessive swapping observed] 12914 2016-06-30
15:06:16.545000 -07:00

2 ORA 445 13161 2016-06-30 15:10:53.756000 -07:00

ADR Home = /scratch/app/crsusr/diag/asm/+asm/+ASM1:

```

```

PROBLEM_ID PROBLEM_KEY LAST_INCIDENT LASTINC_TIME
-----
-----

```

```
1 ORA 445 1177 2016-06-30 15:10:12.930000 -07:00
```

Example 18-3 Show Packages

```
$ tfactl ips show package
```

```
Multiple ADR homepaths were found for /scratch/app/crsusr, please select one ...
```

```
( ) option[0] diag/asmtool/user_root/host_622665046_106
( ) option[1] diag/asmtool/user_crsusr/host_622665046_106
( ) option[2] diag/clients/user_root/host_622665046_106
( ) option[3] diag/clients/user_crsusr/host_622665046_106
( ) option[4] diag/afdbboot/user_root/host_622665046_106
( ) option[5] diag/rdbms/_mgmtdb/-MGMTDB
option[6] Done
```

```
Pls select a homepath [6] ?5
```

```
diag/rdbms/_mgmtdb/-MGMTDB was selected
```

```
PACKAGE_ID          1
PACKAGE_NAME        ORA700kge_20160731211334
PACKAGE_DESCRIPTION
DRIVING_PROBLEM     2
DRIVING_PROBLEM_KEY ORA 700 [kgerev1]
DRIVING_INCIDENT    42605
DRIVING_INCIDENT_TIME 2016-07-05 07:53:28.578000 -07:00
STATUS              Generated (4)
CORRELATION_LEVEL   Typical (2)
PROBLEMS            2 main problems, 0 correlated problems
INCIDENTS          2 main incidents, 0 correlated incidents
INCLUDED_FILES      84

PACKAGE_ID          2
PACKAGE_NAME        IPSPKG_20160801203518
PACKAGE_DESCRIPTION
DRIVING_PROBLEM     N/A
DRIVING_PROBLEM_KEY N/A
DRIVING_INCIDENT    N/A
DRIVING_INCIDENT_TIME N/A
STATUS              Generated (4)
CORRELATION_LEVEL   Typical (2)
PROBLEMS            0 main problems, 0 correlated problems
INCIDENTS          0 main incidents, 0 correlated incidents
INCLUDED_FILES      27
```

Example 18-4 IPS Collect

```
$ tfactl diagcollect -ips
```

```
Collecting data for the last 12 hours for this component ...
```

```
Collecting data for all nodes
```

```
Creating ips package in master node ...
```

```
Multiple ADR homepaths were found for /scratch/app/crsusr, please select one or more...
```

```
( ) option[0] diag/asmtool/user_root/host_622665046_106
( ) option[1] diag/asmtool/user_crsusr/host_622665046_106
( ) option[2] diag/clients/user_root/host_622665046_106
```

```
( ) option[3] diag/clients/user_crsusr/host_622665046_106
( ) option[4] diag/afdbboot/user_root/host_622665046_106
( ) option[5] diag/rdbms/_mgmtdb/-MGMTDB
option[6] Done
```

```
Pls select a homopath [6] ?5
diag/rdbms/_mgmtdb/-MGMTDB was selected
```

Please select at least one ADR homopath.

Multiple ADR homopaths were found for /scratch/app/crsusr, please select one or more...

```
( ) option[0] diag/asmttool/user_root/host_622665046_106
( ) option[1] diag/asmttool/user_crsusr/host_622665046_106
( ) option[2] diag/clients/user_root/host_622665046_106
( ) option[3] diag/clients/user_crsusr/host_622665046_106
( ) option[4] diag/afdbboot/user_root/host_622665046_106
(*) option[5] diag/rdbms/_mgmtdb/-MGMTDB
option[6] Done
```

```
Pls select a homopath [6] ?
Trying ADR basepath /scratch/app/crsusr
Trying to use ADR homopath diag/rdbms/_mgmtdb/-MGMTDB ...
Submitting request to generate package for ADR homopath /scratch/app/crsusr/diag/
rdbms/_mgmtdb/-MGMTDB
Master package completed for ADR homopath /scratch/app/crsusr/diag/rdbms/_mgmtdb/-
MGMTDB
Created package 15 based on time range 2016-08-21 15:58:00.000000 -07:00 to
2016-08-22 03:58:00.000000 -07:00,
correlation level basic
Remote package completed for ADR homopath(s) /diag/rdbms/_mgmtdb/-MGMTDB
```

Collection Id : 20160822035856myserver65

```
Detailed Logging at : /scratch/app/crsusr/tfa/repository/
collection_Mon_Aug_22_03_58_56_PDT_2016_node_all/
diagcollect_20160822035856_myserver65.log
2016/08/22 03:59:40 PDT : Collection Name : tfa_Mon_Aug_22_03_58_56_PDT_2016.zip
2016/08/22 03:59:40 PDT : Collecting diagnostics from hosts : [myserver65,
myserver66]
2016/08/22 03:59:40 PDT : Getting list of files satisfying time range [08/21/2016
15:59:40 PDT, 08/22/2016 03:59:40 PDT]
2016/08/22 03:59:40 PDT : Collecting additional diagnostic information...
2016/08/22 03:59:51 PDT : Completed collection of additional diagnostic
information...
2016/08/22 03:59:51 PDT : Completed Local Collection
2016/08/22 03:59:51 PDT : Remote Collection in Progress...
```

```
-----+-----+-----+-----+
|                Collection Summary                |
+-----+-----+-----+-----+
| Host      | Status   | Size   | Time   |
+-----+-----+-----+-----+
| myserver66 | Completed | 254kB | 16s   |
| myserver65 | Completed | 492kB | 11s   |
+-----+-----+-----+-----+
```

```
Logs are being collected to: /scratch/app/crsusr/tfa/repository/
collection_Mon_Aug_22_03_58_56_PDT_2016_node_all
/scratch/app/crsusr/tfa/repository/collection_Mon_Aug_22_03_58_56_PDT_2016_node_all/
```

```
myserver66.tfa_Mon_Aug_22_03_58_56_PDT_2016.zip  
/scratch/app/crsusr/tfa/repository/collection_Mon_Aug_22_03_58_56_PDT_2016_node_all/  
myserver65.tfa_Mon_Aug_22_03_58_56_PDT_2016.zip
```

Related Topics

- [tfactl ips](#)
Use the `tfactl ips` command to collect Automatic Diagnostic Repository diagnostic data.

19

Managing and Configuring Oracle Trace File Analyzer

This section helps you manage Oracle Trace File Analyzer daemon, diagnostic collections, and the collection repository.

- [Querying Oracle Trace File Analyzer Status and Configuration](#)
Use the `print` command to query the status or configuration.
- [Managing the Oracle Trace File Analyzer Daemon](#)
Oracle Trace File Analyzer runs from `init` on UNIX systems or `init/upstart/systemd` on Linux, or Microsoft Windows uses a Windows Service so that Oracle Trace File Analyzer starts automatically whenever a node starts.
- [Managing the Repository](#)
Oracle Trace File Analyzer stores all diagnostic collections in the repository.
- [Managing Collections](#)
Manage directories configured in Oracle Trace File Analyzer and diagnostic collections.
- [Configuring the Host](#)
You must have `root` or `sudo` access to `tfactl` to add hosts to Oracle Trace File Analyzer configuration.
- [Configuring the Ports](#)
The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005.
- [Configuring SSL and SSL Certificates](#)
View and restrict SSL/TLS protocols. Configure Oracle Trace File Analyzer to use self-signed or CA-signed certificate.

19.1 Querying Oracle Trace File Analyzer Status and Configuration

Use the `print` command to query the status or configuration.

Table 19-1 Configuration Listing and Descriptions

Configuration Listing	Default Value	Description
Automatic diagnostic collection	ON	Triggers a collection if a significant problem occurs. Possible values: <ul style="list-style-type: none">• ON• OFF

Table 19-1 (Cont.) Configuration Listing and Descriptions

Configuration Listing	Default Value	Description
Trimming of files during diagnostic collection	ON	Trims the log files to only entries within the time range of the collection. Possible values: <ul style="list-style-type: none"> • ON • OFF
Repository maximum size in MB	Smaller of either 10GB or 50% of free space in the file system.	The largest size the repository can be.
Trace Level	1	Increases the level of verbosity. Possible values: <ul style="list-style-type: none"> • 1 • 2 • 3 • 4 A value of 1 results in the least amount of trace. A value of 4 results in the most amount of trace. Oracle recommends changing the trace level value only at the request of Oracle Support.
Automatic Purging	ON	Purges collections when: Free space in the repository falls below 1GB. Or Before closing the repository. Purging removes collections from largest size through to smallest. Purging continues until the repository has enough space to open.
Minimum Age of Collections to Purge (Hours)	12	The least number of hours to keep a collection, after which it is eligible for purging.
Minimum Space free to enable Alert Log Scan (MB)	500	Suspends log scanning if free space in the <code>tfa_home</code> falls below this value.

Example 19-1 Print Configuration

```
$ tfactl print config
-----
|                               node1                               |
+-----+-----+-----+-----+
| Configuration Parameter          | Value          |
+-----+-----+-----+-----+
| TFA Version                      | 12.2.1.0.0    |
| Java Version                     | 1.8            |
| Public IP Network                 | true          |
| Automatic Diagnostic Collection   | true          |
| Alert Log Scan                    | true          |
| Disk Usage Monitor                | true          |
```

Managelogs Auto Purge	false
Trimming of files during diagcollection	true
Inventory Trace level	1
Collection Trace level	1
Scan Trace level	1
Other Trace level	1
Repository current size (MB)	447
Repository maximum size (MB)	10240
Max Size of TFA Log (MB)	50
Max Number of TFA Logs	10
Max Size of Core File (MB)	20
Max Collection Size of Core Files (MB)	200
Minimum Free Space to enable Alert Log Scan (MB)	500
Time interval between consecutive Disk Usage Snapshot(minutes)	60
Time interval between consecutive Managelogs Auto Purge(minutes)	60
Logs older than the time period will be auto purged(days[d] hours[h] 30d)	
Automatic Purging	true
Age of Purging Collections (Hours)	12
TFA IPS Pool Size	5

Related Topics

- [tfactl print](#)
Use the `tfactl print` command to print information from the Berkeley database.

19.2 Managing the Oracle Trace File Analyzer Daemon

Oracle Trace File Analyzer runs from `init` on UNIX systems or `init/upstart/systemd` on Linux, or Microsoft Windows uses a Windows Service so that Oracle Trace File Analyzer starts automatically whenever a node starts.

To manage Oracle Trace File Analyzer daemon:

The `init` control file `/etc/init.d/init.tfa` is platform dependant.

1. To start or stop Oracle Trace File Analyzer manually:

- `tfactl start`: Starts the Oracle Trace File Analyzer daemon
- `tfactl stop`: Stops the Oracle Trace File Analyzer daemon

If the Oracle Trace File Analyzer daemon fails, then the operating system restarts the daemon automatically.

2. To enable or disable automatic restarting of the Oracle Trace File Analyzer daemon:

- `tfactl disable`: Disables automatic restarting of the Oracle Trace File Analyzer daemon.
- `tfactl enable`: Enables automatic restarting of the Oracle Trace File Analyzer daemon.

19.3 Managing the Repository

Oracle Trace File Analyzer stores all diagnostic collections in the repository.

The repository size is the maximum space Oracle Trace File Analyzer is able to use on disk to store collections.

- [Purging the Repository Automatically](#)
- [Purging the Repository Manually](#)

19.3.1 Purging the Repository Automatically

Oracle Trace File Analyzer closes the repository, if:

- Free space in `TFA_HOME` is less than 100 MB, also stops indexing
- Free space in `ORACLE_BASE` is less than 100 MB, also stops indexing
- Free space in the repository is less than 1 GB
- Current size of the repository is greater than the repository max size (`resizeMB`)

The Oracle Trace File Analyzer daemon monitors and automatically purges the repository when the free space falls below 1 GB or before closing the repository. Purging removes collections from largest size through to smallest until the repository has enough space to open.

Oracle Trace File Analyzer automatically purges only the collections that are older than `minagetopurge`. By default, `minagetopurge` is 12 hours.

To purge the repository automatically

1. To change the minimum age to purge:

```
set minagetopurge=number of hours
```

For example:

```
$ tfactl set minagetopurge=48
```

Purging the repository automatically is enabled by default.

2. To disable or enable automatic purging:

```
set autopurge=ON|OFF
```

For example:

```
$ tfactl set autopurge=ON
```

3. To change the location of the repository:

```
set repositorydir=dir
```

For example:

```
$ tfactl set repositorydir=/opt/mypath
```

4. To change the size of the repository:

```
set resizeMB
```

For example:

```
$ tfactl set resizeMB=20480
```

Related Topics

- [tfactl set](#)
Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

19.3.2 Purging the Repository Manually

To purge the repository manually:

1. To view the status of the Oracle Trace File Analyzer repository:

```
tfactl print repository
```

2. To view statistics about collections:

```
tfactl print collections
```

3. To manually purge collections that are older than a specific time:

```
tfactl purge -older number[h|d] [-force]
```

Related Topics

- [tfactl purge](#)
Use the `tfactl purge` command to delete diagnostic collections from the Oracle Trace File Analyzer repository that are older than a specific time.
- [tfactl print](#)
Use the `tfactl print` command to print information from the Berkeley database.

19.4 Managing Collections

Manage directories configured in Oracle Trace File Analyzer and diagnostic collections.

- [Including Directories](#)
Add directories to the Oracle Trace File Analyzer configuration to include the directories in diagnostic collections.
- [Managing the Size of Collections](#)
Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and `diagcollect -nocores` to reduce the size of collections.

19.4.1 Including Directories

Add directories to the Oracle Trace File Analyzer configuration to include the directories in diagnostic collections.

Oracle Trace File Analyzer then stores diagnostic collection metadata about the:

- Directory
- Subdirectories
- Files in the directory and all sub directories

All Oracle Trace File Analyzer users can add directories they have read access to.

To manage directories:**1. To view the current directories configured in Oracle Trace File Analyzer**

```
tfactl print directories [ -node all | local | n1,n2,... ]  
[ -comp component_name1,component_name2,.. ]  
[ -policy exclusions | noexclusions ]  
[ -permission public | private ]
```

2. To add directories:

```
tfactl directory add dir  
[ -public ]  
[ -exclusions | -noexclusions | -collectall ]  
[ -node all | n1,n2,... ]
```

3. To remove a directory from being collected:

```
tfactl directory remove dir [ -node all | n1,n2,... ]
```

Related Topics

- [tfactl directory](#)
Use the `tfactl directory` command to add a directory to, or remove a directory from the list of directories to analyze their trace or log files.
- [tfactl print](#)
Use the `tfactl print` command to print information from the Berkeley database.

19.4.2 Managing the Size of Collections

Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and `diagcollect -nocores` to reduce the size of collections.

To manage the size of collections:**1. To trim files during diagnostic collection:**

```
tfactl set trimfiles=ON|OFF
```

- When set to ON (default), Oracle Trace File Analyzer trims files to include data around the time of the event
- When set to OFF, any file that was written to at the time of the event is collected in its entirety

2. To set the maximum size of core file to *n* MB (default 20 MB):

```
tfactl set maxcorefilesize=n
```

Oracle Trace File Analyzer skips core files that are greater than `maxcorefilesize`.

3. To set the maximum collection size of core files to *n* MB (default 200 MB):

```
tfactl set maxcorecollectionsize=n
```

Oracle Trace File Analyzer skips collecting core files after `maxcorecollectionsize` is reached.

4. To prevent the collection of core files with diagnostic collections:

```
tfactl diagcollect -nocores
```

Related Topics

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.
- [tfactl set](#)
Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

19.5 Configuring the Host

You must have `root` or `sudo` access to `tfactl` to add hosts to Oracle Trace File Analyzer configuration.

To add, remove, and replace SSL certificates:

1. To view the list of current hosts in the Oracle Trace File Analyzer configuration:

```
tfactl print hosts
```

2. To add a host to the Oracle Trace File Analyzer configuration for the first time:

- a. If necessary, install and start Oracle Trace File Analyzer on the new host.
- b. From the existing host, synchronize authentication certificates for all hosts by running:

```
tfactl syncnodes
```

If needed, then Oracle Trace File Analyzer displays the current node list it is aware of and prompts you to update this node list.

- c. Select **Y**, and then enter the name of the new host.

Oracle Trace File Analyzer contacts Oracle Trace File Analyzer on the new host to synchronize certificates and add each other to their respective hosts lists.

3. To remove a host:

```
tfactl host remove host
```

4. To add a host and the certificates that are already synchronized:

```
tfactl host add host
```

Oracle Trace File Analyzer generates self-signed SSL certificates during install. Replace those certificates with one of the following:

- Personal self-signed certificate
- CA-signed certificate

19.6 Configuring the Ports

The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005.

If the port range is not available on your system, then replace it with the ports available on your system.

The `$TFA_HOME/internal/usableports.txt` file looks as follows:

```
$ cat $TFA_HOME/internal/usableports.txt
5000
5001
5002
5003
5004
5005
```

To change the ports:

1. Stop Oracle Trace File Analyzer on all nodes:

```
tfactl stop
```
2. Edit the `usableports.txt` file to replace the ports.
3. Replicate the `usableports.txt` changes to all cluster nodes.
4. Remove the `$TFA_HOME/internal/port.txt` file on all nodes.
5. Start Oracle Trace File Analyzer on all nodes:

```
tfactl start
```

19.7 Configuring SSL and SSL Certificates

View and restrict SSL/TLS protocols. Configure Oracle Trace File Analyzer to use self-signed or CA-signed certificate.

- [Configuring SSL/TLS Protocols](#)
The Oracle Trace File Analyzer daemons in a cluster communicate securely using the SSL/TLS protocols.
- [Configuring Self-Signed Certificates](#)
Use `Java keytool` to replace self-signed SSL certificates with personal self-signed certificates.
- [Configuring CA-Signed Certificates](#)
Use `Java keytool` and `openssl` to replace self-signed SSL certificates with the Certificate Authority (CA) signed certificates.
- [Configuring SSL Cipher Suite](#)
The cipher suite is a set of cryptographic algorithms used by the TLS/SSL protocols to create keys and encrypt data.

19.7.1 Configuring SSL/TLS Protocols

The Oracle Trace File Analyzer daemons in a cluster communicate securely using the SSL/TLS protocols.

The SSL protocols available for use by Oracle Trace File Analyzer are:

- TLSv1.2
- TLSv1.1
- TLSv1

Oracle Trace File Analyzer always restricts use of older the protocols `SSLv3` and `SSLv2Hello`.

7. Restrict the permissions on the keystores to `root read-only`.

```
$ chmod 400 myclient.jks myserver.jks
```
8. Copy the keystores (`jks` files) to each node.
9. Configure Oracle Trace File Analyzer to use the new certificates:

```
$ tfactl set sslconfig
```
10. Restart the Oracle Trace File Analyzer process to start using new certificates:

```
$ tfactl stop
$ tfactl start
```

19.7.3 Configuring CA-Signed Certificates

Use Java `keytool` and `openssl` to replace self-signed SSL certificates with the Certificate Authority (CA) signed certificates.

To configure Oracle Trace File Analyzer to use CA-signed certificates:

1. Create a private key for the server request:

```
$ openssl genrsa -aes256 -out myserver.key 2048
```
2. Create a private key for the client request:

```
$ openssl genrsa -aes256 -out myclient.key 2048
```
3. Create a Certificate Signing Request (CSR) for the server:

```
$ openssl req -key myserver.key -new -sha256 -out myserver.csr
```
4. Create a Certificate Signing Request (CSR) for the client:

```
$ openssl req -key myclient.key -new -sha256 -out myclient.csr
```
5. Send the resulting CSR for the client and the server to the relevant signing authority.

The signing authority sends back the signed certificates:

- `myserver.cert`
- `myclient.cert`
- CA root certificate

6. Convert the certificates to JKS format for the server and the client:

```
$ openssl pkcs12 -export -out serverCert.pkcs12 -in myserver.cert -inkey
myserver.key
```

```
$ keytool -v -importkeystore -srckeystore serverCert.pkcs12 -srcstoretype PKCS12
-destkeystore myserver.jks -deststoretype JKS
```

```
$ openssl pkcs12 -export -out clientCert.pkcs12 -in myclient.cert -inkey
myclient.key
```

```
$ keytool -v -importkeystore -srckeystore clientCert.pkcs12 -srcstoretype PKCS12
-destkeystore myclient.jks -deststoretype JKS
```

7. Import the server public key into to the client `jks` file:

```
$ keytool -import -v -alias server-ca -file myserver.cert -keystore myclient.jks
```

8. Import the client public key to the server `jks` file:

```
$ keytool -import -v -alias client-ca -file myclient.cert -keystore myserver.jks
```

9. Import the CA root certificate from the signing authority into the Oracle Trace File Analyzer server certificate:

```
$ keytool -importcert -trustcacerts -alias inter -file caroot.cert -keystore myserver.jks
```

10. Restrict the permissions on the keystores to `root` `read-only`:

```
$ chmod 400 myclient.jks myserver.jks
```

11. Copy the keystores (`jks` files) to each node.

12. Configure Oracle Trace File Analyzer to use the new certificates:

```
$ tfactl set sslconfig
```

13. Restart the Oracle Trace File Analyzer process to start using the new certificates.

```
$ tfactl stop  
$ tfactl start
```

19.7.4 Configuring SSL Cipher Suite

The cipher suite is a set of cryptographic algorithms used by the TLS/SSL protocols to create keys and encrypt data.

Oracle Trace File Analyzer supports any of the cipher suites used by JRE 1.8.

The default cipher suite used is `TLS_RSA_WITH_AES_128_CBC_SHA256`.

1. You can change the cipher suite with the command:

```
tfactl set ciphersuite=cipher_suite
```

For example:

```
tfactl set ciphersuite=TLS_RSA_WITH_AES_128_GCM_SHA256
```

For a list of JRE cipher suites, see:

<https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunJSSEProvider>

Oracle Trace File Analyzer Service

Oracle Trace File Analyzer Service is deployed as part of Domain Service Cluster (DSC) setup.

- [Overview of Oracle Trace File Analyzer Service](#)
Oracle Trace File Analyzer service helps DBAs and system administrators proactively monitor trace files across multiple member clusters on a centralized machine.
- [Cluster or Host View](#)
The cluster or host view shows all clusters in a table heatmap, which you can drill-down to the host level.
- [Detailed View](#)
Detailed view displays a dashboard with top events represented in a timeline.
- [File Browser](#)
Oracle Trace File Analyzer Service classifies each trace file and provides an interface to browse files by various classifications.
- [File Viewer](#)
File viewer opens each file in a new tab.
- [Merged Files](#)
To find what's happening in different components at the same time, look at multiple trace files.
- [Importing CA Certificates into Tomcat](#)
Oracle Trace File Analyzer service starts an Apache Tomcat instance to serve web requests.

20.1 Overview of Oracle Trace File Analyzer Service

Oracle Trace File Analyzer service helps DBAs and system administrators proactively monitor trace files across multiple member clusters on a centralized machine.

Oracle Trace File Analyzer helps the DBAs and system administrators to analyze the data for issues, identify duplicate issues, and predict future failures.

Oracle Trace File Analyzer Service is set up as part of Domain Service Cluster (DSC) setup. You need not run any commands to configure or set up Oracle Trace File Analyzer Service. The service runs on the first node of a cluster.

To print details about Oracle Trace File Analyzer Service running in Domain Service Cluster (DSC), run the following command as Oracle Grid Infrastructure user or as root:

```
$ tfactl receiver info
TFA Service URL           : http://rws1270066:7070/tfa/index.html
TFA Service URL (https)   : https://rws1270066:7071/tfa/index.html
TFA Service Admin User    : admin
TFA Service Admin Status  : active
TFA Service Repository    : /scratch/app/oragrid/tfa/repository
```

```
TFA Service Port      : 7001
TFA Service Members  :
```

After the initial setup, web administrator account is locked.

To unlock, run the following command as Oracle Grid Infrastructure user or as root:

```
$ tfactl receiver reset webadmin
```

20.2 Cluster or Host View

The cluster or host view shows all clusters in a table heatmap, which you can drill-down to the host level.

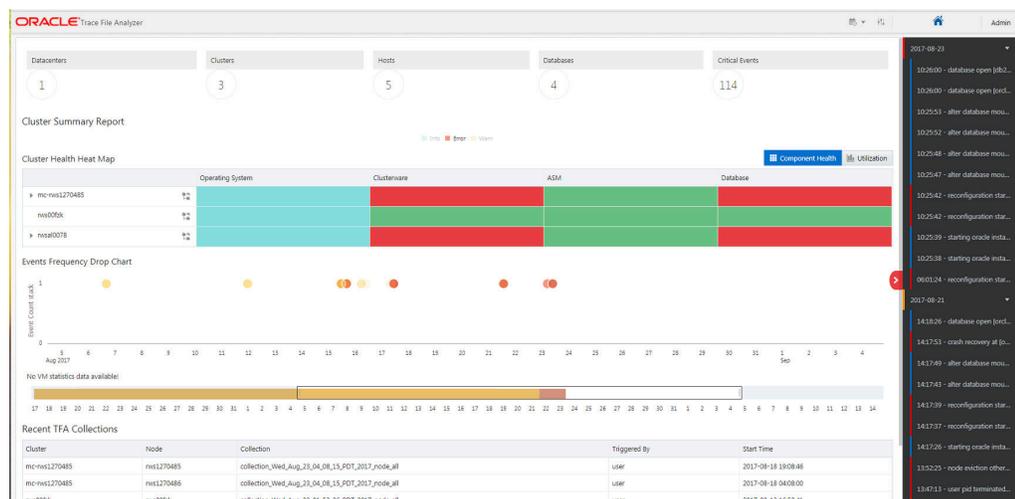
Clicking on a cluster shows:

- Configuration of that cluster
- Status
- ORACLE_HOMES
- Version
- Patch level for each home
- Databases running from home and their statuses

The y-axis represents cluster or host name and the x-axis represents the health of important components in the cluster.

Metrics view shows the CPU, Memory, Disk usage, and Critical Events from each host on x-axis.

By clicking on a cell, you can drill-down details for the selected host or component.

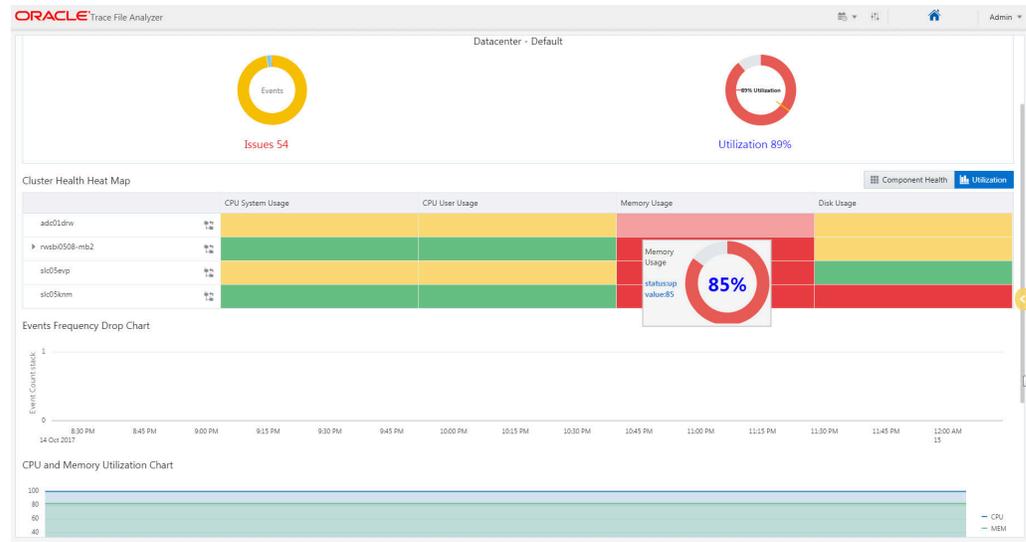


Clicking on Utilization shows system utilization broken down by:

- CPU system usage
- CPU user usage

- Memory usage
- Disk usage

Hovering on a section of the heatmap shows more information.



20.3 Detailed View

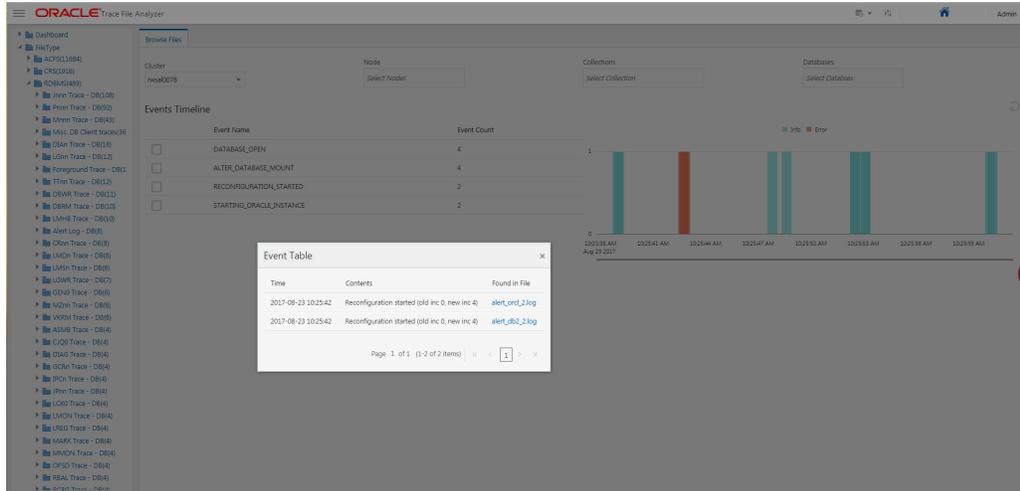
Detailed view displays a dashboard with top events represented in a timeline.

Click any event to view a short analysis including:

- Files where event is seen
- Events around the time
- CPU and memory usage around the time

Click a file to open:

- Browse files by file type, folder, collection
- View files
- Merge multiple files



20.4 File Browser

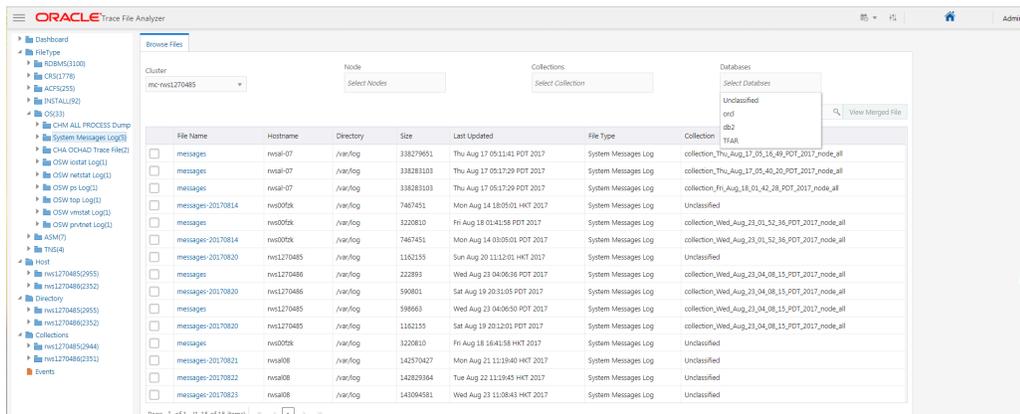
Oracle Trace File Analyzer Service classifies each trace file and provides an interface to browse files by various classifications.

Classifications make it easy to find important files across thousands of trace files generated by various components.

The interface shows a tree navigation menu on left to browse files by type, host name, actual directory, or by Oracle Trace File Analyzer collections. You can also filter the file list by cluster, node, or Oracle Trace File Analyzer collection. There is a search option to find file names by keyword search.

Use the tree on the left to click browse files by file type, folder, and collection.

Click a file to open it or merge multiple files to view combined events across multiple files.

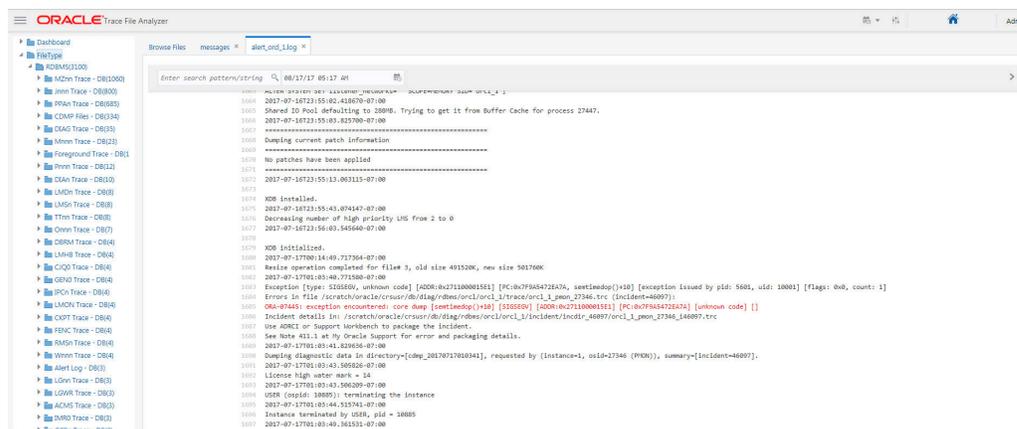


20.5 File Viewer

File viewer opens each file in a new tab.

You can move back and forth during the analysis of a problem without losing the context.

Within a file, file viewer highlights important lines to identify the critical lines instantly. You can also search within a file and jump to a specific section.

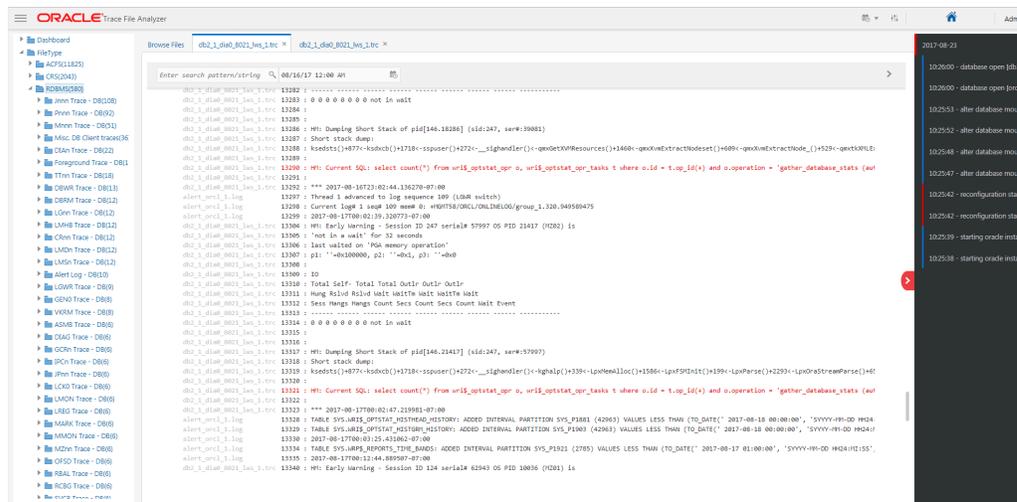


20.6 Merged Files

To find what's happening in different components at the same time, look at multiple trace files.

The merge feature combines multiple timeline trace files by time and provides a single view.

This sample output shows a database alert log combined with system messages file.



20.7 Importing CA Certificates into Tomcat

Oracle Trace File Analyzer service starts an Apache Tomcat instance to serve web requests.

- Tomcat is set up at `$TFA_HOME/tomcat`.
- Tomcat configuration file: `$TFA_HOME/tomcat/conf/server.xml`.



Note:

For detailed instructions to set up SSL, refer to the official Tomcat documentation:

[SSL/TLS Configuration HOW-TO](#)

To import CA Certificates into Tomcat:

1. Stop the web server.

```
$CRS_HOME/bin/tfactl receiver stop
```

2. Create a keystore and import the custom or CA certificate.

For more details, refer to the official tomcat documentation.

3. Edit Tomcat configuration.

Look for SSL/TLS configuration section as shown. Update the keystore/certificate information.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
This connector uses the NIO implementation that requires the JSSE
style configuration. When using the APR/native implementation, the
OpenSSL style configuration is required as described in the APR/native
documentation -->
<Connector port="7071" protocol="org.apache.coyote.http11.Http11NioProtocol"
maxThreads="150" SSLEnabled="true" scheme="https" secure="true"
keystoreFile="<existing file>" keystorePass="<existing keystorePass>"
clientAuth="false" sslProtocol="TLS" />
```

Please refer to tomcat documentation for different configuration options.

4. Start the web server.

```
$CRS_HOME/bin/tfactl receiver start
```

21

Managing Oracle Database and Oracle Grid Infrastructure Diagnostic Data

This section enables you to manage Oracle Database and Oracle Grid Infrastructure diagnostic data and disk usage snapshots.

- [Managing Automatic Diagnostic Repository Log and Trace Files](#)
Use the `managelogs` command to manage Automatic Diagnostic Repository log and trace files.
- [Managing Disk Usage Snapshots](#)
Use `tfactl` commands to manage Oracle Trace File Analyzer disk usage snapshots.
- [Purging Oracle Trace File Analyzer Logs Automatically](#)
Use these `tfactl` commands to manage log file purge policy for Oracle Trace Analyzer log files.

21.1 Managing Automatic Diagnostic Repository Log and Trace Files

Use the `managelogs` command to manage Automatic Diagnostic Repository log and trace files.

The `-purge` command option removes files managed by Automatic Diagnostic Repository. This command clears files from "ALERT", "INCIDENT", "TRACE", "CDUMP", "HM", "UTSCDMP", "LOG" under diagnostic destinations. The `-purge` command also provides details about the change in the file system space.

If the diagnostic destinations contain large numbers of files, then the command runs for a while. Check the removal of files in progress from the corresponding directories.

To remove files, you must have operating system privileges over the corresponding diagnostic destinations.

To manage Automatic Diagnostic Repository log and trace files:

1. To limit purge, or show operations to only files older than a specific time:

```
$ tfactl managelogs -older mm|h|d Files from past 'n' [d]ays or 'n' [h]ours or 'n' [m]inutes
```

For example:

```
$ tfactl managelogs -purge -older 30d -dryrun
```

```
$ tfactl managelogs -purge -older 30d
```

2. To get an estimate of how many files are removed and how much space is freed, use the `-dryrun` option:

For example:

```
$ tfactl managelogs -purge -older 30d -dryrun
```

3. To remove files and clean disk space:

For example:

```
$ tfactl managelogs -purge -older 30d
```

```
$ tfactl managelogs -purge -older 30d -gi
```

```
$ tfactl managelogs -purge -older 30d -database
```

4. To view the space usage of individual diagnostic destinations:

For example:

```
$ tfactl managelogs -show usage
```

```
$ tfactl managelogs -show usage -gi
```

```
$ tfactl managelogs -show usage -database
```

Related Topics

- [tfactl managelogs](#)
Use the `tfactl managelogs` command to manage Automatic Diagnostic Repository log and trace files.

21.2 Managing Disk Usage Snapshots

Use `tfactl` commands to manage Oracle Trace File Analyzer disk usage snapshots.

Oracle Trace File Analyzer automatically monitors disk usage, records snapshots, and stores the snapshots under `tfa/repository/suptools/node/managelogs/usage_snapshot/`.

By default, the time interval between snapshots is 60 minutes.

To manage disk usage snapshots:

1. To change the default time interval for snapshots:

```
$ tfactl set diskUsageMonInterval=minutes
```

where *minutes* is the number of minutes between snapshots.

2. To turn the disk usage monitor on or off:

```
$ tfactl set diskUsageMon=ON|OFF
```

21.3 Purging Oracle Trace File Analyzer Logs Automatically

Use these `tfactl` commands to manage log file purge policy for Oracle Trace Analyzer log files.

Automatic purging is enabled by default on a Domain Service Cluster (DSC), and disabled by default elsewhere. When automatic purging is enabled, every 60 minutes, Oracle Trace File Analyzer automatically purges logs that are older than 30 days.

To purge Oracle Trace File Analyzer logs automatically:

1. To turn on or off automatic purging:

```
$ tfactl set manageLogsAutoPurge=ON|OFF
```

2. To adjust the age of logs to purge:

```
$ tfactl set manageLogsAutoPurgePolicyAge=nd|h
```

3. To adjust the frequency of purging:

```
$ tfactl set manageLogsAutoPurgeInterval=minutes
```

22

Troubleshooting Oracle Trace File Analyzer

This section helps you diagnose and remediate Oracle Trace File Analyzer issues.

- [Cluster Nodes are Not Showing As One Cluster When Viewed by Running the tfactl status Command](#)
- [Oracle Trace File Analyzer is Not Starting and the init.tfa script is Missing After Reboot](#)
- [Error Message Similar to "Can't locate **** in @inc \(@inc contains:....\)"](#)
- [Non-Release Update Revisions \(RURs\) Oracle Trace File Analyzer Patching Fails on Remote Nodes](#)
- [Non-Root Access is Not Enabled After Installation](#)
- [TFA_HOME and Repository Locations are Moved After Patching or Upgrade](#)
- [Oracle Trace File Analyzer Fails with TFA-00103 After Applying the July 2015 Release Update Revision \(RUR\) or Later](#)
- [OSWatcher Parameters are Different After a Reboot or Otherwise Unexpectedly Different](#)
- [Oracle Trace File Analyzer Installation or Oracle Trace File Analyzer Discovery \(tfactl rediscover\) Fails on Linux 7](#)
- [OSWatcher Analyzer Fails When OSWatcher is Not Running from the TFA_HOME](#)
- [Oracle Trace File Analyzer Fails to Start with com.sleepycat.je.EnvironmentLockedException Java Exception](#)
- [Oracle Trace File Analyzer Startup Fails When Solution-Soft Time Machine Software is Installed, but Not Running on the System](#)
- [Non-privileged User is Not Able to Run tfactl Commands?](#)
- [Oracle Trace File Analyzer Daemon is Not Starting or Not Running?](#)

22.1 Cluster Nodes are Not Showing As One Cluster When Viewed by Running the tfactl status Command

Cause: Certificates are not synchronized.

Action: Manually synchronize the keys.

Go to any one of the cluster nodes and run the `synctfanodes.sh` script as `root`.

```
# $GIHOME/tfa/nodename/tfa_home/bin/synctfanodes.sh
```

 **Note:**

The script uses SSH and SCP. If passwordless SSH is not set for `root`, then Oracle Trace File Analyzer prompts you 3 times per node for password each time a command is run.

If the Expect utility is available on the node, then Oracle Trace File Analyzer uses Expect thus reducing the number of prompts for password.

22.2 Oracle Trace File Analyzer is Not Starting and the init.tfa script is Missing After Reboot

Description: The file system housing `TFA_HOME` with Oracle Trace File Analyzer binaries was not mounted when `init.tfa` was run from `init` or `System D` on Linux 6 and above.

Cause: There are many reasons and not restricted to the following:

- Mounting the file system was disabled for maintenance or patching
- Problems or errors related to the file system
- NFS inaccessible network
- File system with `TFA_HOME` is mounting slowly

Action: Refer to My Oracle Support note 2224163.1 to fix this issue.

Related Topics

- <https://support.oracle.com/rs?type=doc&id=2224163.1>

22.3 Error Message Similar to "Can't locate **** in @inc (@inc contains:....)"

Cause: Using an old version of Perl causes this error.

Action: Oracle Trace File Analyzer requires Perl version 5.10 or above. If you encounter similar errors, then upgrade Perl to version 5.10 or above.

After installing, update the location of Perl in the `tfa_home/tfa_setup.txt` file to point to the new location:

```
PERL=/u01/perl/bin/perl
```

If the problem occurs during install, then use the `-perlhome dir` install option.

The directory you specify must contain `/bin/perl`. If you install Perl as `root`, then `root` must own the Perl executable.

```
# which perl
/usr/bin/perl
```

```
# ./installTFA-LINUX -perlhome /usr
```

22.4 Non-Release Update Revisions (RURs) Oracle Trace File Analyzer Patching Fails on Remote Nodes

Cause: Remote nodes fail to upgrade due to a socket issue when upgrading Oracle Trace File Analyzer through Oracle Trace File Analyzer sockets.

Description: After completing the upgrade, crosscheck the report if all nodes are at the same version, build id, and status.

Host	TFA Version	TFA Build ID	Upgrade Status
node1	12.1.2.6.0	12126020151019114604	UPGRADED
node2	12.1.2.6.0	12126020151019114604	UPGRADED

If you see any differences as follows, then you must fix the issue.

Host	TFA Version	TFA Build ID	Upgrade Status
node1	12.1.2.6.0	12126020151019114604	UPGRADED
node2	12.1.2.3.0	12120020140619094932	NOT UPGRADED

Action: Copy the Oracle Trace File Analyzer installer to all nodes that failed to upgrade and run the installer locally on those nodes.

```
./installTFALite -local
```

After upgrading the binaries, replace the root SSL certificates from the node that initiated upgrade.

Copy the following files from the exiting configuration node to the node to be added. Change the permission for those files to 700 for root on the machine to be added.

```
tfa_home/server.jks
tfa_home/client.jks
tfa_home/internal/ssl.properties
```

22.5 Non-Root Access is Not Enabled After Installation

Description: Non-root access for the Oracle Grid Infrastructure software owner must be activated by default when non-root access is enabled.

Action: To enable non-root access to Oracle Trace File Analyzer, run the `tfactl access add -user command as root`.

For example:

```
tfactl access add -user xyx
```

Running command enables the non-root user group `xyx` to access Oracle Trace File Analyzer.

22.6 TFA_HOME and Repository Locations are Moved After Patching or Upgrade

Description: Before Oracle Trace File Analyzer version 12.1.2.6.0, when an existing free standing Oracle Trace File Analyzer is installed (MOS version installed outside the `GRID_HOME`) and Oracle Trace File Analyzer is then patched with Oracle Grid Infrastructure as part of Oracle 12.1.0.2, then `TFA_HOME` is moved into the `GRID_HOME` and the repository directory is moved to the Oracle Grid Infrastructure owners Oracle base directory.

If the repository directory is changed to a non-default location, then the change is lost.

- To set the Oracle Trace File Analyzer zip file repository location to the required base directory, run the `tfactl set repositorydir` command.
- To change the maximum size of the Oracle Trace File Analyzer repository, run the `tfactl set resizeMB` command.

Starting with Oracle Trace File Analyzer version 12.1.2.6.0 and above, if `TFA_HOME` exists outside the `GRID_HOME`, then Oracle Trace File Analyzer installation is moved as part of Release Update Revision (RUR) installation. However, if the Release Update Revision (RUR) has a newer version of Oracle Trace File Analyzer, then Oracle Trace File Analyzer is upgraded in its current location.

If Oracle Trace File Analyzer is installed in the `GRID_HOME` and the `GRID_HOME` is moved as part of any patching, then the existing `TFA_HOME` is migrated to the new `GRID_HOME` and upgraded as required.

22.7 Oracle Trace File Analyzer Fails with TFA-00103 After Applying the July 2015 Release Update Revision (RUR) or Later

- [Phase 1 of Oracle Trace File Analyzer upgrade](#)
- [Phase 2 of Oracle Trace File Analyzer upgrade](#)
- [How can I verify that both phases have been completed and that Oracle Trace File Analyzer communication among all the nodes has been established?](#)
- [What if I do not upgrade all my nodes at the same time by choice or if some are down for maintenance?](#)
- [I know that not all nodes are upgraded at the same time. I do not want to wait 24 hours for Oracle Trace File Analyzer to sync the key files. What do I do?](#)

Phase 1 of Oracle Trace File Analyzer upgrade

Oracle Trace File Analyzer communication model has been changed in versions greater than 12.1.2.4.1. To avoid communication problems, Oracle Trace File Analyzer communication change must be complete across all nodes of the Oracle Trace File Analyzer configuration. Oracle Trace File Analyzer is upgraded on each node locally as part of application of Release Update Revision (RUR). The Release Update

Revision (RUR) process applies the new software and restarts Oracle Trace File Analyzer, but does not put in place the new connection model.

Phase 2 of Oracle Trace File Analyzer upgrade

Before automatically implementing the new communication model, Oracle Trace File Analyzer waits for 24 hours to complete the application of Release Update Revision (RUR) on all nodes. Once Oracle Trace File Analyzer is upgraded on all the nodes, phase 2 must occur within 10 minutes. The new Oracle Trace File Analyzer communication model is not implemented (phase 2) until Release Update Revision (RUR) is applied on all nodes (phase 1).

Oracle Trace File Analyzer indicates by displaying the message:

```
TFA-00103 - TFA is not yet secured to run all commands.
```

Once Oracle Trace File Analyzer is upgraded on all nodes in the configuration (phase 1), Oracle Trace File Analyzer:

- Generates new SSL keys
- Sends the keys to the valid nodes in the cluster
- Restart Oracle Trace File Analyzer on each of these nodes (phase 2)

On completion of phase 2, Oracle Trace File Analyzer must process commands normally using the new communication model.

How can I verify that both phases have been completed and that Oracle Trace File Analyzer communication among all the nodes has been established?

First, as `root` run:

```
tfactl print status
```

```

-----
| Host      | Status  | PID    | Port  | Version | Build ID          | Inventory |
-----+-----+-----+-----+-----+-----+-----
| sales1   | RUNNING | 4390   | 5000 | 12.1.2.4.2 | 12124220150629072212 | COMPLETE |
| sales2   | RUNNING | 23604  | 5000 | 12.1.2.4.2 | 12124220150629072212 | COMPLETE |
| sales3   | RUNNING | 28653  | 5000 | 12.1.2.4.2 | 12124220150629072212 | COMPLETE |
| sales4   | RUNNING | 5989   | 5000 | 12.1.2.4.2 | 12124220150629072212 | COMPLETE |
-----+-----+-----+-----+-----+-----+-----

```

Once all nodes are shown to be at the same version and build ID then within about 10 minutes maximum the synchronization of keys must complete.

Ensure that you run the following command:

```
tfactl print directories
```

Running `tfactl print directories` must return the list of directories registered in Oracle Trace File Analyzer. If the communication is not established among all the nodes, then the command returns the message, TFA is not yet secured to run all commands.

The message also indicates that phase 2 has not been completed. To verify on which nodes phase 2 has not yet been completed, on each node, check the existence of the following files. The files must be readable only by `root`, ownership:group of `root`. The checksum for each file must match on all nodes.

```
# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/client.jks
-rwx----- 1 root root 3199 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/client.jks

# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/server.jks
-rwx----- 1 root root 3201 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/server.jks

# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/internal/ssl.properties
-rwx----- 1 root root 220 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/internal/ssl.properties
```

What if I do not upgrade all my nodes at the same time by choice or if some are down for maintenance?

Oracle Trace File Analyzer waits to complete the phase 2 operations until all nodes have completed upgrade or until 24 hours has passed.

After 24 hours, Oracle Trace File Analyzer:

- Generates new keys
- Copies the key to all the nodes that have been upgraded
- Restarts Oracle Trace File Analyzer on those nodes

Any nodes that did not get the keys are outside of the Oracle Trace File Analyzer configuration. After upgrading Oracle Trace File Analyzer, manually synchronize the keys with other nodes.

If the application of Release Update Revision (RUR) on all the nodes is completed within 24 hours, then manually synchronize the keys.

To manually synchronize the keys, go to one node that has completed Phase 2 and run the `synctfanodes.sh` script as `root`.

```
# $GIHOME/tfa/nodename/tfa_home/bin/synctfanodes.sh
```



Note:

The script uses SSH and SCP. If `root` does not have passwordless SSH, then Oracle Trace File Analyzer prompts you 3 time per node for password each time a command is run.

If the Expect utility is available on the node, then Oracle Trace File Analyzer uses Expect thus reducing the number of prompts for password.

The script displays all the nodes in Oracle Trace File Analyzer configuration, including the nodes where Oracle Trace File Analyzer is yet to upgrade.

The script also shows the nodes that are part of the Oracle Grid Infrastructure configuration.

Verify the node list provided and supply a space-separated list of nodes to synchronize. It doesn't hurt to include the nodes that were previously upgraded as the process is idempotent.

For example:

Nodes *sales1*, *sales2*, *sales3*, and *sales4* are all part of Oracle Grid Infrastructure. The nodes were running Oracle Trace File Analyzer 12.1.2.0.0 until the July 2015 Release Update Revision (RUR) was applied.

The Release Update Revision (RUR) was applied initially only to *sales1* and *sales3* due to outage restrictions.

After completion of phase 1 of the Oracle Trace File Analyzer upgrade, run `print status`. Running the command lists all nodes even though different versions of Oracle Trace File Analyzer are running on some of the nodes.

```
-bash-3.2# /u01/app/12.1.0/grid/bin/tfactl print status
```

Host	Status	PID	Port	Version	Build ID	Inventory
sales1	RUNNING	27270	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales3	RUNNING	19222	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales2	RUNNING	10141	5000	12.1.2.0.0	12120020140619094932	COMPLETE
sales4	RUNNING	17725	5000	12.1.2.0.0	12120020140619094932	COMPLETE

Since the new Oracle Trace File Analyzer communication model is not set up among all the nodes, many commands when run as `root` fail with the message:

```
TFA is not yet secured to run all commands.
```

Failed attempts to run `tfactl` commands as a non-root indicates that there is no sufficient permission to use Oracle Trace File Analyzer.

After 24 hours, Oracle Trace File Analyzer completes phase 2 for *sales1* and *sales3*. Oracle Trace File Analyzer communication model is established for *sales1* and *sales3*. You can perform normal Oracle Trace File Analyzer operations on *sales1* and *sales3*. Communication with *sales2* and *sales4* has not yet been established and so running remote commands to them fail.

When running `print status` on *sales1* and *sales3*, we no longer see *sales2* and *sales4*. Only Oracle Trace File Analyzer using the new Oracle Trace File Analyzer communication model communicates.

```
-bash-3.2# /u01/app/12.1.0/grid/bin/tfactl print status
```

Host	Status	PID	Port	Version	Build ID	Inventory
sales1	RUNNING	4390	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales3	RUNNING	23604	5000	12.1.2.4.2	12124220150629072212	COMPLETE

Running the command `tfactl diagcollect` collects from *sales1* and *sales3* but not from the other nodes.

```
-bash-3.2$ /u01/app/12.1.0/grid/bin/tfactl diagcollect
Collecting data for the last 4 hours for this component...
Collecting data for all nodes
```

```

Repository Location in sales1 : /u01/app/oragrid/tfa/repository
2015/06/30 05:25:27 PDT : Collection Name : tfa_Tue_Jun_30_05_25_20_PDT_2015.zip
2015/06/30 05:25:27 PDT : Sending diagcollect request to host : sales2
2015/06/30 05:25:27 PDT : Sending diagcollect request to host : sales3
2015/06/30 05:25:27 PDT : Sending diagcollect request to host : sales4
2015/06/30 05:25:27 PDT : Scanning of files for Collection in progress...
....
....
....
2015/06/30 05:25:37 PDT : Remote Collection in Progress...
2015/06/30 05:25:57 PDT : sales3:Completed Collection
2015/06/30 05:26:07 PDT : sales2:Failed Unable to connect to Node sales2
2015/06/30 05:26:07 PDT : sales4:Failed Unable to connect to Node sales4
2015/06/30 05:26:07 PDT : Completed collection of zip files.

```

While upgrading on the remaining nodes, Oracle Trace File Analyzer cannot see the nodes already upgraded until the configuration is synchronized.

```
bash-3.2# /u01/app/12.1.0/grid/bin/tfactl print status
```

```

-----
| Host   | Status | PID | Port | Version   | Build ID           | Inventory |
+-----+-----+-----+-----+-----+-----+-----+
| sales3 | RUNNING | 9   | 5000 | 12.1.2.4.2 | 12124220150629072212 | COMPLETE |
+-----+-----+-----+-----+-----+-----+-----+

```

For nodes, on which the application of Release Update Revision (RUR) was not completed within the 24 hour waiting period to become part of Oracle Trace File Analyzer configuration:

1. Run the synchronize script from a node that has the keys already generated
2. Manually copy the SSL configuration to those nodes

In our example from *sales1*:

```
/u01/app/12.1.0/grid/tfa/sales1/tfa_home/bin/synctfanodes.sh
```

```
Current Node List in TFA :
```

```
sales1
sales2
sales3
sales4
```

```
Node List in Cluster :
```

```
sales1 sales2 sales3 sales4
```

```
Node List to sync TFA Certificates :
```

```
1 sales2
2 sales3
3 sales4
```

```
Do you want to update this node list? [Y|N] [N]: Y
```

```
Please Enter all the nodes you want to sync...
```

```
Enter Node List (seperated by space) : sales2 sales4
```

```
Syncing TFA Certificates on sales2 :
```

```
TFA_HOME on sales2 : /u01/app/12.1.0/grid/tfa/sales2/tfa_home
```

```
Copying TFA Certificates to sales2...
```

```

Copying SSL Properties to sales2...
Shutting down TFA on sales2...
Sleeping for 5 seconds...
Starting TFA on sales2...

Syncing TFA Certificates on sales4 :

TFA_HOME on sales4 : /u01/app/12.1.0/grid/tfa/sales4/tfa_home

Copying TFA Certificates to sales4...
Copying SSL Properties to sales4...
Shutting down TFA on sales4...
Sleeping for 5 seconds...
Starting TFA on sales4...

Successfully re-started TFA..

```

Host	Status	PID	Port	Version	Build ID	Inventory
sales1	RUNNING	4390	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales2	RUNNING	23604	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales3	RUNNING	28653	5000	12.1.2.4.2	12124220150629072212	COMPLETE
sales4	RUNNING	5989	5000	12.1.2.4.2	12124220150629072212	COMPLETE

 **Note:**

The node list was changed to only the nodes that needed the keys synchronized, *sales2* and *sales4*.

In this case, it's fine to synchronize *sales3* as it would have received the same files and restart Oracle Trace File Analyzer.

I know that not all nodes are upgraded at the same time. I do not want to wait 24 hours for Oracle Trace File Analyzer to sync the key files. What do I do?

Use the synchronize script to force Oracle Trace File Analyzer to generate and synchronize certificates. While running, the script prompts if you wish to generate SSL configuration files and then synchronizes them to the remote nodes.

For example:

```

-bash-3.2# /u01/app/12.1.0/grid/tfa/sales1/tfa_home/bin/synctfanodes.sh

Current Node List in TFA :
sales1
sales2
sales3
sales4

TFA has not yet generated any certificates on this Node.

Do you want to generate new certificates to synchronize across the nodes? [Y|N] [Y]:

Generating new TFA Certificates...

Restarting TFA on sales1...

```

```

Shutting down TFA
TFA-00002 : Oracle Trace File Analyzer (TFA) is not running
TFA Stopped Successfully
. . . . .
. . .
Successfully shutdown TFA..
Starting TFA..
Waiting up to 100 seconds for TFA to be started..
. . . . .
. . . . .
Successfully started TFA Process..
. . . . .
TFA Started and listening for commands

Node List in Cluster :
sales1 sales2 sales3 sales4

Node List to sync TFA Certificates :
1 sales2
2 sales3
3 sales4

Do you want to update this node list? [Y|N] [N]:

```

After the key files are generated and synchronized, on each node you must find the files as follows:

```

# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/client.jks

-rwx----- 1 root    root      3199 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/client.jks

# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/server.jks

-rwx----- 1 root    root      3201 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/server.jks

# ls -al /u01/app/12.1.0/grid/tfa/sales1/tfa_home/internal/ssl.properties

-rwx----- 1 root    root      220 Jun 30 14:12 /u01/app/12.1.0/grid/tfa/sales1/
tfa_home/internal/ssl.properties

```

Readable only by root, ownership:group of root. The checksum for each file must match on all nodes.

22.8 OSWatcher Parameters are Different After a Reboot or Otherwise Unexpectedly Different

When Oracle Trace File Analyzer manages OSWatcher, after an install or a reboot, OSWatcher is started as a non-privileged user such as:

- grid on Oracle RAC systems
- oracle on non-Oracle RAC systems

Oracle does not recommend stopping and restarting OSWatcher as root.

For example:

```
tfactl oswbb stop
```

```
tfactl start oswbb 20 72 (interval of 20 seconds and retention of 72 hours)
```

OSWatcher is then run as `root` until it is stopped and re-started as `oracle` or `grid`, or there is a reboot. In either case, the parameters are persisted in a property file. OSWatcher defaults (30,48) are used unless other parameters are specified for interval and retention period. Beginning with Oracle Trace File Analyzer version 12.1.2.5.2, an OSWatcher property file is maintained for each user. Each time OSWatcher is started, the parameters for interval or retention hours are made persistent for that user. In earlier versions, if the OSWatcher startup parameters are different than expected, then it is because OSWatcher was stopped and started as `root` with different parameters. These settings would have persisted across reboots because there was only one properties file.

In 12.1.2.5.2 and above, if there is a reboot, then OSWatcher must always be brought up using the parameters from the properties of `oracle` or `grid`. The OSWatcher startup parameters are different if OSWatcher is stopped and re-started as `root` with different parameters before a reboot. The parameters fetched from the `root` properties must not take effect after a reboot. The parameters must revert to the parameters of `oracle` properties.

The parameters are different and the persistent settings are changed because Oracle Support would have recommended different settings to investigate an issue. In that case, stop, and re-start OSWatcher with the normal parameters as a non-privileged user.

```
tfactl oswbb stop
```

```
tfactl start oswbb (in this case the default interval of 30 seconds and retention of 48 hours would be persisted)
```

Note:

If OSWatcher is installed and running, and not managed by Oracle Trace File Analyzer, then Oracle Trace File Analyzer defers to that installation and parameters. When listing the `oswbb` tool status, the status must be **NOT RUNNING**, that is, not managed by Oracle Trace File Analyzer.

22.9 Oracle Trace File Analyzer Installation or Oracle Trace File Analyzer Discovery (tfactl rediscover) Fails on Linux 7

Description: Reported errors are similar to:

```
Can't locate Data/Dumper.pm in @INC (@INC contains: /usr/local/lib64/perl5
/usr/local/share/perl5 /usr/lib64/perl5/vendor_perl
/usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .
/u01/app/12.1.0/grid/tfa/dc75orac02/tfa_home/bin
/u01/app/12.1.0/grid/tfa/dc75orac02/tfa_home/bin/common
/u01/app/12.1.0/grid/tfa/dc75orac02/tfa_home/bin/modules
/u01/app/12.1.0/grid/tfa/dc75orac02/tfa_home/bin/common/exceptions) at
/u01/app/12.1.0/grid/tfa/dc75orac02/tfa_home/bin/common/tfactlshare.pm line 545.
```

Cause: This error occurs due to Bug 21790910 and Bug 22393355, which are fixed in Oracle Trace File Analyzer version 12.1.2.6.4.

Action: Link the operating system Perl to the version of Perl in the `GRID_HOME`.

22.10 OSWatcher Analyzer Fails When OSWatcher is Not Running from the TFA_HOME

Description: Reported errors are similar to:

```
tfactl> oswbb
Error: Cannot find OSWatcher files under
/u01/app/grid/tfa/repository/suptools//oswbb//archive
OSWatcher analyzer commands are supported only when it is running from TFA_HOME
```

Cause: Expected behavior when OSWatcher is not running from `TFA_HOME`.

Action:

1. Stop and disable the OSWatcher version running outside of Oracle Trace File Analyzer.
2. Start OSWatcher from within Oracle Trace File Analyzer.

22.11 Oracle Trace File Analyzer Fails to Start with `com.sleepycat.je.EnvironmentLockedException` Java Exception

Description: Reported errors found in the Oracle Trace File Analyzer `syserrorout` log located in `$TFA_BASE//log` are:

```
/u01/app/oracle/tfa//log$ cat syserrorout.08.06.2015-16.19.54

Exception in thread "TFAMain" com.sleepycat.je.EnvironmentLockedException: (JE
5.0.84)
/u01/app/oracle/tfa//database/BERKELEY_JE_DB The environment cannot be locked for
single writer access.
ENV_LOCKED: The je.lck file could not be locked. Environment is invalid and must be
closed.
at com.sleepycat.je.log.FileManager.(FileManager.java:368)
at com.sleepycat.je.dbi.EnvironmentImpl.(EnvironmentImpl.java:483)
at com.sleepycat.je.dbi.EnvironmentImpl.(EnvironmentImpl.java:409
```

Cause: The root cause is unknown.

Action:

1. Check if there are any processes accessing the BDB.


```
# fuser $GI_BASE/tfa//database/BERKELEY_JE_DB/je.lck
```
2. If a process is returned, then kill it.


```
# kill -9
```

3. Remove the `$GI_BASE/tfa//database/BERKELEY_JE_DB/je.lck` file.

```
# rm -rf $GI_BASE/tfa//database/BERKELEY_JE_DB/je.lck
```

4. Start Oracle Trace File Analyzer.

```
# $TFA_HOME/bin/tfactl start
```

22.12 Oracle Trace File Analyzer Startup Fails When Solution-Soft Time Machine Software is Installed, but Not Running on the System

Action: Uninstall the Time Machine software.

22.13 Non-privileged User is Not Able to Run tfactl Commands?

Description:

As `root` verify that the non-privileged user has Oracle Trace File Analyzer privilege to run the `tfactl` commands.

```
tfactl access lsuser
/u01/app/12.1.0/grid/bin/tfactl access lsusers
-----
| TFA Users in myNode1 |
+-----+-----+-----+
| User Name | User Type | Status |
+-----+-----+-----+
| oracle   | USER    | Allowed |
+-----+-----+-----+
```

If the user is listed and the status is displayed as **Disabled**, then that indicates all non-privileged user access has been disabled.

Action:

To enable non-privileged user access:

```
tfactl access enable
```

If the user, for example, `oracle` is not listed, then add `oracle`.

```
tfactl access add -user oracle
```

If none of the above techniques resolve the problem, then run `tfactl diagnosetfa -local`. Upload the resultant file to Oracle Support.

22.14 Oracle Trace File Analyzer Daemon is Not Starting or Not Running?

Description:

TFA-00001: Failed to start Oracle Trace File Analyzer (TFA) daemon

TFA-00002: Oracle Trace File Analyzer (TFA) is not running

The errors indicate that Java does not start.

Action:

1. Verify that Oracle Trace File Analyzer is not running.

```
ps -ef|grep -i tfa
```

Note:

On some operating systems, the `ps` command truncates the output at 80 characters. The `ps` command does not display the process even if it is running.

2. To confirm that the Oracle Trace File Analyzer daemon is not running, run the following command run as `root`.

```
# tfactl print status
```

3. Try starting the Oracle Trace File Analyzer daemon as `root`.

```
# tfactl start
```

If Oracle Trace File Analyzer still fails to start, then run `tfactl diagnosetfa -local`. Upload the resultant file to Oracle Support.

Part VI

Appendixes

- [Oracle ORAchk and Oracle EXAchk Command-Line Options](#)
Most command-line options apply to both Oracle ORAchk and Oracle EXAchk. Use the command options to control the behavior of Oracle ORAchk and Oracle EXAchk.
- [OCLUMON Command Reference](#)
Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.
- [Managing the Cluster Resource Activity Log](#)
Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.
- [chactl Command Reference](#)
The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.
- [Oracle Trace File Analyzer Command-Line and Shell Options](#)
The Trace File Analyzer control utility, TFACTL, is the command-line interface for Oracle Trace File Analyzer.

Oracle ORAchk and Oracle EXAchk Command-Line Options

Most command-line options apply to both Oracle ORAchk and Oracle EXAchk. Use the command options to control the behavior of Oracle ORAchk and Oracle EXAchk.

Syntax

```
$ ./orachk options

[-h] [-a] [-b] [-v] [-p] [-m] [-u] [-f] [-o]
[-clusternodes clusternames]
[-failedchecks previous_result]
[-nordbms]
[-output path]
[-dbnames dbnames]
[-localonly]
[-debug]
[-dbnone | -dball]
[-c]
[-upgrade | -noupgrade]
[-syslog]
[-skip_usr_def_checks]
[-checkfaileduploads]
[-uploadfailed all | comma-delimited list of collections]
[-fileattr [start | check | remove] [-includedir path] [-excludediscovery] [-baseline path] [-fileattronly]
```

```

[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-setdbupload all | db upload variable, for example, RAT_UPLOAD_CONNECT_STRING,
RAT_UPLOAD_PASSWORD]
[-unsetdbupload all | db upload variable, for example, RAT_UPLOAD_CONNECT_STRING,
RAT_UPLOAD_PASSWORD]
[-checkdbupload]
[-getdbupload]
[-cmupgrade]
[-sendemail "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-nopass]
[-noscore]
[-showpass]
[-show_critical]
[-diff Old Report New Report [-outfile Output HTML] [-force]]
[-merge report 1 report 2 [-force]]
[-tag tagname]
[-nodaemon]
[-profile asm | clusterware | corroborate | dba | ebs | emagent | emoms | em |
goldengate | hardware | maa | oam | oim | oud | ovn | peoplesoft | preinstall |
prepatch | security | siebel | solaris_cluster | storage | switch | sysadmin |
timesten | user_defined_checks | zfs ]
[-excludeprofile asm | clusterware | corroborate | dba | ebs | emagent | emoms |
em | goldengate | hardware | maa | oam | oim | oud | ovn | peoplesoft | preinstall
| prepatch | security | siebel | solaris_cluster | storage | switch | sysadmin |
timesten | user_defined_checks | zfs ]
[-acchk -javahome path to jdk8
-asmhome path to asm-all-5.0.3.jar -appjar directory where jar files are present for
concrete class -apptrc directory where trace files are present for coverage class]
[-check check ids | -excludecheck check ids]
[-zfsnodes nodes]
[-zfssa appliance names]
[-dbserial | -dbparallel [n] | -dbparallelmax]
[-idmpreinstall | -idmpostinstall | -idmruntime] [-topology topology.xml |
-credconfig credconfig] | -idmdbpreinstall | -idmdbpostinstall | -idmbruntime]
[-idm_config IDMCONFIG] [-idmdiscargs IDMDISCARGS]
[-idmhcargs IDMHCARGS | -h]

```

\$./exachk options

```

[-h] [-a] [-b] [-v] [-p] [-m] [-u] [-f] [-o]
[-clusternodes clusternames]
[-output path]
[-dbnames dbnames]
[-localonly]
[-debug]
[-dbnone | -dball]
[-c]
[-upgrade | -noupgrade]
[-syslog] [-skip_usr_def_checks]
[-checkfaileduploads]
[-uploadfailed all | comma-delimited list of collections]
[-fileattr start | check | remove [-includedir path [-excludediscovery] [-baseline
path[-fileattronly]
[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-setdbupload all | db upload variable, for example, RAT_UPLOAD_CONNECT_STRING,
RAT_UPLOAD_PASSWORD]
[-unsetdbupload all | db upload variable, for example, RAT_UPLOAD_CONNECT_STRING,
RAT_UPLOAD_PASSWORD]
[-checkdbupload]
[-getdbupload]
[-cmupgrade] [-sendemail "NOTIFICATION_EMAIL=comma-delimited list of email

```

```

addresses" ]
[-nopass]
[-noscore]
[-showpass]
[-show_critical]
[-diff Old Report New Report [-outfile Output HTML] [-force]]
[-merge report 1 report 2 [-force]]
[-tag tagname]
[-auto_restart -initsetup | -initdebugsetup | -initrmsetup | -initcheck | -h]
[-d start|start_debug|stop|status|info|stop_client|nextautorun|-h]
[-nodaemon]
[-unlockcells all | -cells comma-delimited list of names or IPs of cells] [-lockcells all | -cells comma-delimited list of names or IPs of cells]
[-usecompute]
[-exadiff Exalogic collection1 Exalogic collection2]
[-vmguest ]
[-hybrid [-phy nodes]]
[-profile asm | bi_middleware | clusterware | compute_node | exatier1 | control_VM | corroborate | dba | ebs | el_extensive | el_lite | el_rackcompare | emagent | emoms | em | goldengate | hardware | maa | nimbula | obiee | ovn | peoplesoft | platinum | preinstall | prepatch | security | siebel | solaris_cluster | storage | switch | sysadmin | timesten | user_defined_checks | virtual_infra]
[-excludeprofile asm | bi_middleware | clusterware | compute_node | exatier1 | control_VM | corroborate | dba | ebs | el_extensive | el_lite | el_rackcompare | emagent | emoms | em | goldengate | hardware | maa | nimbula | obiee | ovn | peoplesoft | platinum | preinstall | prepatch | security | siebel | solaris_cluster | storage | switch | sysadmin | timesten | user_defined_checks | virtual_infra]
[-check check ids | -excludecheck check ids]
[-cells cells]
[-ibswitches switches]
[-torswitches]
[-extzfsnodes nodes]
[-dbserial | -dbparallel [n] | -dbparallelmax | -allserial]
[-allserial | -dbnodeserial |-cellserial | -switchserial]

```

- [Running Generic Oracle ORAchk and Oracle EXAchk Commands](#)
List of command options common to Oracle ORAchk and Oracle EXAchk.
- [Controlling the Scope of Checks](#)
Use the list of commands in this section to control the scope of checks.
- [Managing the Report Output](#)
Use the list of commands in this section to manage the report output.
- [Uploading Results to Database](#)
Use the list of commands in this section to upload results to the database.
- [Configuring the Daemon Mode](#)
Use the daemon to configure automatic health check runs at scheduled intervals.
- [Controlling the Behavior of the Daemon](#)
Use the list of commands in this section to control the behavior of the daemon.
- [Tracking File Attribute Changes](#)
Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.

Running Generic Oracle ORAchk and Oracle EXAchk Commands

List of command options common to Oracle ORAchk and Oracle EXAchk.

Syntax

```
[-a]
[-v]
[-debug]
[-nodaemon]
[-f]
[-upgrade]
[-noupgrade]
[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-sendemail "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-dbserial]
[-dbparallel [n]]
[-dbparallelmax]
```

Parameters

Table 1 Generic Commands

Option	Description
-a	Runs all checks, including the best practice checks and the recommended patch check. If you do not specify any options, then the tools run all checks by default.
-v	Shows the version of Oracle ORAchk and Oracle EXAchk tools.
-debug	Runs in debug mode. The generated .zip file contains a debug log and other files useful for Oracle Support.
-nodaemon	Does not send commands to the daemon, usage is interactive.
-f	Runs Offline. The tools perform health checks on the data already collected from the system.
-upgrade	Forces an upgrade of the version of the tools being run.
-noupgrade	Does not prompt for an upgrade even if a later version is available under the location specified in the RAT_UPGRADE_LOC environment variable.
-testemail all "NOTIFICATION_EMAIL=co mma-delimited list of email addresses"	Sends a test email to validate email configuration.
-sendemail "NOTIFICATION_EMAIL=co mma-delimited list of email addresses"	Specify a comma-delimited list of email addresses. Emails the generated HTML report on completion to the specified email addresses.
-dbserial	Runs the SQL, SQL_COLLECT, and OS health checks in serial.

Table 1 (Cont.) Generic Commands

Option	Description
-dbparallel <i>[n]</i>	Runs the SQL, SQL_COLLECT, and OS health checks in parallel, using <i>n</i> number of child processes. Default is 25% of CPUs.
-dbparallelmax	Runs the SQL, SQL_COLLECT, and OS health checks in parallel, using the maximum number of child processes.

Controlling the Scope of Checks

Use the list of commands in this section to control the scope of checks.

Syntax

```
[-b]
[-p]
[-m]
[-u -o pre]
[-u -o post]
[-clusternodes nodes]
[-failedchecks previous_result]
[-nordbms]
[-dbnames db_names]
[-dbnone]
[-dball]
[-localonly]
[-cells cells]
[-ibswitches switches]
[-profile profile]
[-excludeprofile profile]
[-check check_id]
[-excludecheck check_id]
[-skip_usr_def_checks]
```

Parameters

Table 2 Scope of Checks

Command	Description
-b	Runs only the best practice checks. Does not run the recommended patch checks.
-p	Runs only the patch checks.
-m	Excludes the checks for Maximum Availability Architecture (MAA) scorecards.
-u -o pre	Runs the pre-upgrade checks for Oracle Clusterware and database.
-u -o post	Runs the post-upgrade checks for Oracle Clusterware and database.
-clusternodes <i>nodes</i>	Specify a comma-delimited list of node names to run only on a subset of nodes.

Table 2 (Cont.) Scope of Checks

Command	Description
<code>-failedchecks previous_result</code>	Runs only checks from the <i>previous_result</i> , which had failed.
<code>-nordbms</code>	Runs Oracle Grid Infrastructure checks only in environments with no Oracle Database checks performed.
<code>-dbnames db_names</code>	Specify a comma-delimited list of database names to run only on a subset of databases.
<code>-dbnone</code>	Does not prompt for database selection and skips all the database checks.
<code>-dball</code>	Does not prompt for database selection and runs the database checks on all databases discovered on the system.
<code>-localonly</code>	Runs only on the local node.
<code>-cells cells</code>	Specify a comma-delimited list of storage server names to run the checks only on a subset of storage servers.
<code>-ibswitches switches</code>	Specify a comma-delimited list of InfiniBand switch names to run the checks only on a subset of InfiniBand switches.
<code>-profile profile</code>	Specify a comma-delimited list of profiles to run only the checks in the specified profiles.
<code>-excludeprofile profile</code>	Specify a comma-delimited list of profiles to exclude the checks in the specified profiles.
<code>-check check_id</code>	Specify a comma-delimited list of check IDs to run only the checks specified in the list check IDs.
<code>-excludecheck check_id</code>	Specify a comma-delimited list of check IDs to exclude the checks specified in the list of check IDs.
<code>-skip_usr_def_checks</code>	Does not run the checks specified in the user-defined xml file.

Managing the Report Output

Use the list of commands in this section to manage the report output.

Syntax

```
[-syslog] [-tag tagname]
[-o]
[-nopass]
[-noscore]
[-diff old_report new_report [-outfile output_HTML]]
[-merge [-force] collections]
```

Parameters

Table 3 Managing Output

Option	Description
<code>-syslog</code>	Writes JSON results to syslog.

Table 3 (Cont.) Managing Output

Option	Description
-tag <i>tagname</i>	Appends the <i>tagname</i> specified to the output report name. The <i>tagname</i> must contain only alphanumeric characters.
-o	Argument to an option. If -o is followed by v, (or <i>verbose</i> , and neither option is case-sensitive), then the command prints passed checks on the screen. If the -o option is not specified, then the command prints only the failed checks on the screen.
-nopass	Does not show passed checks in the generated output.
-noscore	Does not print health score in the HTML report.
-diff <i>old_report</i> <i>new_report</i> [-outfile <i>output_HTML</i>]	Reports the difference between the two HTML reports. Specify a directory name or a ZIP file or an HTML report file as <i>old_report</i> and <i>new_report</i> .
-merge [-force] <i>collections</i>	Merges a comma-delimited list of collections and prepares a single report.

Uploading Results to Database

Use the list of commands in this section to upload results to the database.

Syntax

```
[-setdbupload all|list of variable names]
[-unsetdbupload all|list of variable names]
[-checkdbupload]
[-getdbupload]
[-checkfaileduploads]
[-uploadfailed all|list of failed collections]
```

Parameters

Table 4 Uploading Results to Database

Option	Description
-setdbupload all <i>variable_names</i>	Sets the values in the wallet to upload health check run results to the database. all: Sets all the variables in the wallet. <i>variable_names</i> : Specify a comma-delimited list of variables to set.
-unsetdbupload all <i>variable_names</i>	Unsets the values in the wallet to upload health check run results to the database. all: Unsets all the variables in the wallet. <i>variable_names</i> : Specify a comma-delimited list of variables to unset.
-checkdbupload	Checks if the variables are set correctly for uploading the health check run results to the database.
-getdbupload	Prints the variables with their values from wallet for uploading the health check run result to the database.

Table 4 (Cont.) Uploading Results to Database

Option	Description
-checkfaileduploads	Reports any failed collection uploads.
-uploadfailed all list of failed collections	Reattempts to upload one or more failed collection uploads. all: Reattempts to upload all the failed collection uploads. list of failed collections: Specify a comma-delimited list of collections to upload.

Configuring the Daemon Mode

Use the daemon to configure automatic health check runs at scheduled intervals.



Note:

Daemon mode is supported only on the Linux and Solaris operating systems.



Note:

If you have an Oracle Engineered System, then in addition to the following usage steps, follow the system-specific instructions.

1. Set the daemon properties.

At a minimum, set `AUTORUN_SCHEDULE` and `NOTIFICATION_EMAIL`.

For example, to set the tool to run at 3 AM every Sunday and email the results to `some.body@example.com`, run the following command:

```
$ ./orachk -set "AUTORUN_SCHEDULE=3 * *  
0 ;NOTIFICATION_EMAIL=some.body@example.com"
```

```
$ ./exachk -set "AUTORUN_SCHEDULE=3 * *  
0 ;NOTIFICATION_EMAIL=some.body@example.com"
```

2. Configure the health check daemon.
3. Start the daemon as `root` (recommended) or as the Oracle Database or Oracle Grid Infrastructure home owner.

```
# ./orachk -d start
```

```
# ./exachk -d start
```

4. Answer the questions prompted during startup.

Controlling the Behavior of the Daemon

Use the list of commands in this section to control the behavior of the daemon.

Syntax

```
[-id id] -set daemon_option  
[-id id] -unset daemon_option | all  
[-id id] -get parameter | all  
[-d start]  
[-d start -debug]  
[-d stop]  
[-d stop_client]  
[-d status]  
[-d info]  
[-id id] -d nextautorun  
[-initsetup]  
[-initrmsetup]  
[-initcheck]
```

Parameters

Table 5 Daemon Options

Option	Description
[-id id] -set daemon_option	Optionally use <i>id</i> with the <code>set</code> command to set specific daemon usage profiles.
[-id id] -unset daemon_option all	Unsets the parameter. Use with <code>-id id</code> to set a daemon profile-specific value.
[-id id] -get parameter all	Displays the value of the specified parameter or all the parameters. Use with <code>-id id</code> to set a daemon profile-specific value.
-d start	Starts the daemon.
-d start -debug	Starts the daemon in debug mode.
-d stop	Stops the daemon.
-d stop_client	Forces a running daemon client to stop.
-d status	Checks the current status of the daemon.
-d info	Displays details about the daemon. The details include installation and when the daemon was started.
[-id id] -d nextautorun	Displays details about when the next scheduled automatic run occurs.
-initsetup	Sets the daemon auto restart function that starts the daemon when the node starts.
-initrmsetup	Removes the automatic restart functionality.
-initcheck	Checks if the automatic restart functionality is set up.

Tracking File Attribute Changes

Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.

Syntax

```
[-fileattr start]  
[-fileattr check]
```

```
[-fileattr remove]
[-fileattr [start|check] -includedir directories]
[-fileattr [start|check] -excludediscovery]
[-fileattr check -baseline baseline snapshot path]
[-fileattr check -fileattronly]
```

Table 6 List of Oracle ORAchk and Oracle EXAchk File Attribute Tracking Options

Option	Description
-fileattr start	Takes file attribute snapshots of discovered directories, and stores the snapshots in the output directory. By default, this option takes snapshots of Oracle Grid Infrastructure homes, and all the installed Oracle Database homes. If a user does not own a particular directory, then the tool does not take snapshots of the directory.
-fileattr check	Takes a new snapshot of discovered directories, and compares it with the previous snapshot.
-fileattr remove	Removes file attribute snapshots and related files.
-fileattr [start check] -includedir <i>directories</i>	Specify a comma-delimited list of directories to check file attributes. For example: ./orachk -fileattr start -includedir "/root/home,/etc" ./orachk -fileattr check -includedir "/root/home,/etc"
-fileattr [start check] -excludediscovery	Excludes the discovered directories. For example: ./orachk -fileattr start -includedir "/root/home,/etc" -excludediscovery
-fileattr check -baseline <i>baseline snapshot path</i>	Uses a snapshot that you designate as the baseline for a snapshot comparison. Provide the path to the snapshot that you want to use as the baseline. A baseline is the starting file attributes that you want to compare to at later times. Current file attributes are compared to the baseline and a delta is reported. For example: ./orachk -fileattr check -baseline "/tmp/Snapshot"
-fileattr check -fileattronly	Performs only file attributes check, and then exits Oracle ORAchk. For example: ./orachk -fileattr check -fileattronly

OCLUMON Command Reference

Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.

Use OCLUMON to perform miscellaneous administrative tasks, such as changing the debug levels, querying the version of Cluster Health Monitor, and changing the metrics database size.

- [oclumon debug](#)
Use the `oclumon debug` command to set the log level for the Cluster Health Monitor services.
- [oclumon dumpnodeview](#)
Use the `oclumon dumpnodeview` command to view log information from the system monitor service in the form of a node view.
- [oclumon manage](#)
Use the `oclumon manage` command to view and change configuration information from the system monitor service.
- [oclumon version](#)
Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.

oclumon debug

Use the `oclumon debug` command to set the log level for the Cluster Health Monitor services.

Syntax

```
oclumon debug [log daemon module:log_level] [version]
```

Parameters

Table 7 oclumon debug Command Parameters

Parameter	Description
<code>log daemon module:log_level</code>	<p>Use this option change the log level of daemons and daemon modules.</p> <p>Supported daemons are:</p> <ul style="list-style-type: none"><code>osysmond</code><code>ologgerd</code><code>client</code><code>all</code> <p>Supported daemon modules are:</p> <ul style="list-style-type: none"><code>osysmond</code>: CRFMOND, CRFM, and allcomp<code>ologgerd</code>: CRFLOGD, CRFLDREP, CRFM, and allcomp<code>client</code>: OCLUMON, CRFM, and allcomp<code>all</code>: allcomp <p>Supported <code>log_level</code> values are 0, 1, 2, and 3.</p>
<code>version</code>	<p>Use this option to display the versions of the daemons.</p>

Example 1 oclumon debug

The following example sets the log level of the system monitor service (`osysmond`):

```
$ oclumon debug log osysmond CRFMOND:3
```

The following example displays the versions of the daemons:

```
$ oclumon debug version

OCLUMON version :0.02
OSYSMOND version :12.01
OLOGGERD version :2.01
NODEVIEW version :12.01
Clusterware version - label date:
    12.2.0.1.0 - 160825
```

oclumon dumpnodeview

Use the `oclumon dumpnodeview` command to view log information from the system monitor service in the form of a node view.

Usage Notes

A node view is a collection of all metrics collected by Cluster Health Monitor for a node at a point in time. Cluster Health Monitor attempts to collect metrics every five seconds on every node. Some metrics are static while other metrics are dynamic.

A node view consists of eight views when you display verbose output:

- **SYSTEM:** Lists system metrics such as CPU COUNT, CPU USAGE, and MEM USAGE
- **TOP CONSUMERS:** Lists the top consuming processes in the following format:
metric_name: 'process_name(process_identifier) utilization'
- **CPUS:** Lists statistics for each CPU
- **PROCESSES:** Lists process metrics such as PID, name, number of threads, memory usage, and number of file descriptors
- **DEVICES:** Lists device metrics such as disk read and write rates, queue length, and wait time per I/O
- **NICS:** Lists network interface card metrics such as network receive and send rates, effective bandwidth, and error rates
- **FILESYSTEMS:** Lists file system metrics, such as total, used, and available space
- **PROTOCOL ERRORS:** Lists any protocol errors

Generate a summary report that only contains the SYSTEM and TOP CONSUMERS views.

Syntax

```
oclumon dumpnodeview [-allnodes | -n node1 ...] [-last duration | -s timestamp -e timestamp] [-i interval] [-v | [-system][-process][-procag][-device][-filesystem][-nic][-protoerr][-cpu][-topconsumer]] [-format format type] [-dir directory [-append]]
```

Parameters

Table 8 oclumon dumpnodeview Command Parameters

Parameter	Description
-allnodes	Use this option to dump the node views of all the nodes in the cluster.
-n <i>node1 node2</i>	Specify one node or several nodes in a space-delimited list for which you want to dump the node view.
-last " <i>duration</i> "	Use this option to specify a time, given in HH24:MM:SS format surrounded by double quotation marks (" "), to retrieve the last metrics. For example: "23:05:00"
-s " <i>time_stamp</i> " -e " <i>time_stamp</i> "	Use the -s option to specify a time stamp from which to start a range of queries and use the -e option to specify a time stamp to end the range of queries. Specify time in YYYY-MM-DD HH24:MM:SS format surrounded by double quotation marks (" "). For example: "2011-05-10 23:05:00" Note: Specify these two options together to obtain a range.
-i <i>interval</i>	Specify a collection interval, in five-second increments.
-v	Displays verbose node view output.
-system, -process, -device, -filesystem, -nic, -protoerr, -cpu, -topconsumer	Dumps each specified node view parts.
-format " <i>format type</i> "	Specify the output format. " <i>format type</i> " can be legacy, tabular, or csv. The default format is mostly tabular with legacy for node view parts with only one row.
-dir <i>directory</i>	Dumps the node view to the files in the directory that you specify. Specify the -append option to append the files of the current to the existing files. If you do not specify -append, then the command overwrites the existing files, if present. For example, the command oclumon dumpnodeview -dir <i>dir_name</i> dumps the data in the specified directory. If this command is run twice, it overwrites the data dumped by the previous run. Running the command with -append, for example, oclumon dumpnodeview -dir <i>dir_name</i> -append, appends the data of the current run with the previous one in the specified directory.

Table 8 (Cont.) oclumon dumpnodeview Command Parameters

Parameter	Description
-procag	Outputs the process of the node view, aggregated by category: <ul style="list-style-type: none"> • DBBG (DB backgrounds) • DBFG (DB foregrounds) • CLUST (Cluster) • OTHER (other processes) Note: -procag is currently available only on Linux, Solaris, and AIX. It is not supported on Microsoft Windows systems.
-h	Displays online help for the oclumon dumpnodeview command.

Usage Notes

- In certain circumstances, data can be delayed for some time before the command replays the data.
For example, the `crsctl stop cluster -all` command can cause data delay. After running `crsctl start cluster -all`, it may take several minutes before `oclumon dumpnodeview` shows any data collected during the interval.
- The default is to continuously dump node views. To stop continuous display, use Ctrl+C on Linux and Microsoft Windows.
- Both the local system monitor service (`osysmond`) and the cluster logger service (`ologgerd`) must be running to obtain node view dumps.
- The `oclumon dumpnodeview` command displays only 127 CPUs of the CPU core, omitting a CPU at random from the list.

Metric Descriptions

This section includes descriptions of the metrics in each of the seven views that comprise a node view listed in the following tables.

Table 9 oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
#pcpus	Number of physical CPUs.
#cores	Number of CPU cores in the system.
#vcpus	Number of logical compute units.
cpuht	CPU hyperthreading enabled (Y) or disabled (N).
chipname	Name of the CPU vendor.
cpu	Average CPU utilization per processing unit within the current sample interval (%).

Table 9 (Cont.) oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
	Percentage of over all CPU cores. 100% indicates that all cores are spent for that metric.
cpuusage	Total CPU usage = $cpusystem + cpuuser + cpunice$
cpusystem	CPU used by processes in kernel mode.
cpuuser	CPU used by normal processes in user mode.
cpunice	CPU used by "niced" processes (low priority).
cpuiowait	CPU waiting for I/O.
cpusteal	Virtual CPU waiting for physical CPU to be freed by other VM.
cpuq	Number of processes waiting in the run queue within the current sample interval.
physmemfree	Amount of free RAM (KB).
physmemtotal	Amount of total usable RAM (KB).
shmem	Shared memory.
mcache	Amount of physical RAM used for file buffers plus the amount of physical RAM used as cache memory (KB). On Microsoft Windows systems, this is the number of bytes currently being used by the file system cache. Note: This metric is not available on Solaris.
swapfree	Amount of swap memory free (KB)
swaptotal	Total amount of physical swap memory (KB)
hugepagetotal	Total size of huge in KB Note: This metric is not available on Solaris or Microsoft Windows systems.
hugepagefree	Free size of huge page in KB Note: This metric is not available on Solaris or Microsoft Windows systems.
hugepagesize	Smallest unit size of huge page Note: This metric is not available on Solaris or Microsoft Windows systems.

Table 9 (Cont.) oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
ior	Average total disk read rate within the current sample interval (KB per second).
iow	Average total disk write rate within the current sample interval (KB per second).
ios	Average disk I/O operation rate within the current sample interval (I/O operations per second).
swpin	Average swap in rate within the current sample interval (KB per second). Note: This metric is not available on Microsoft Windows systems.
swpout	Average swap out rate within the current sample interval (KB per second). Note: This metric is not available on Microsoft Windows systems.
pgin	Average page in rate within the current sample interval (pages per second).
pgout	Average page out rate within the current sample interval (pages per second).
netr	Average total network receive rate within the current sample interval (KB per second).
netw	Average total network send rate within the current sample interval (KB per second).
procs	Number of processes.
procsoncpu	The current number of processes running on the CPU.
#procs_blocked	Number of processes currently blocked waiting for I/O.
rtprocs	Number of real-time processes.
rtprocsoncpu	The current number of real-time processes running on the CPU.
#fds	Number of open file descriptors. <i>or</i> Number of open handles on Microsoft Windows.
#sysfdlimit	System limit on the number of file descriptors. Note: This metric is not available on either Solaris or Microsoft Windows systems.
#disks	Number of disks.
#nics	Number of network interface cards.
nicErrors	Average total network error rate within the current sample interval (errors per second).

Table 9 (Cont.) oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
#nfs	Number of network file system.
loadavg1 loadavg5 loadavg15	Load average (average number of jobs in the run queue or waiting for disk I/O) of the last 1, 5, 15 minutes.

Table 10 oclumon dumpnodeview PROCESSES View Metric Descriptions

Metric	Description
name	The name of the process executable.
pid	The process identifier assigned by the operating system.
ppid	PID of the parent process. For example, if process 1 spawns process 2, then ppid of process 2 is pid of process 1.
cumulative_cpu	The total amount of CPU time this process is scheduled to run since it started. The total amount of CPU time spent for this process so far is measured in micro seconds.
#procfdlimit	Limit on number of file descriptors for this process. Note: This metric is not available on Microsoft Windows, AIX, and HP-UX systems.
cpuusage	Process CPU utilization (%). Note: The utilization value can be up to 100 times the number of processing units.
vmem	Process virtual memory usage (KB).
privmem	Process private memory usage (KB).
shmem, shm, and sharedmem	Process shared memory usage (KB). Note: This metric is not available on Microsoft Windows, Solaris, and AIX systems. It is supported only on Linux systems.
workingset	Working set of a program (KB) Note: This metric is only available on Microsoft Windows.
#fd	Number of file descriptors open by this process. <i>or</i> Number of open handles by this process on Microsoft Windows.
#threads	Number of threads created by this process.
priority	The process priority.

Table 10 (Cont.) oclumon dumpnodeview PROCESSES View Metric Descriptions

Metric	Description
nice	The nice value of the process. Note: This metric is not applicable to Microsoft Windows systems.
state	The state of the process. Note: This metric is not applicable to Microsoft Windows systems.

Table 11 oclumon dumpnodeview DEVICES View Metric Descriptions

Metric	Description
ior	Average disk read rate within the current sample interval (KB per second).
iow	Average disk write rate within the current sample interval (KB per second).
ios	Average disk I/O operation rate within the current sample interval (I/O operations per second)
qlen	Number of I/O requests in <code>WAIT</code> state within the current sample interval.
wait	Average wait time per I/O within the current sample interval (msec).
type	If applicable, identifies what the device is used for. Possible values are: <ul style="list-style-type: none"> • SWAP • SYS • OCR • ASM • VOTING

Table 12 oclumon dumpnodeview NICS View Metric Descriptions

Metric	Description
netrr	Average network receive rate within the current sample interval (KB per second).
netwr	Average network sent rate within the current sample interval (KB per second).
neteff	Average effective bandwidth within the current sample interval (KB per second)
nicerrors	Average error rate within the current sample interval (errors per second).
pktsin	Average incoming packet rate within the current sample interval (packets per second).
pktsout	Average outgoing packet rate within the current sample interval (packets per second).

Table 12 (Cont.) oclumon dumpnodeview NICS View Metric Descriptions

Metric	Description
errsin	Average error rate for incoming packets within the current sample interval (errors per second).
errout	Average error rate for outgoing packets within the current sample interval (errors per second).
indiscarded	Average drop rate for incoming packets within the current sample interval (packets per second).
outdiscarded	Average drop rate for outgoing packets within the current sample interval (packets per second).
inunicast	Average packet receive rate for unicast within the current sample interval (packets per second).
type	Whether PUBLIC or PRIVATE.
innonunicast	Average packet receive rate for multi-cast (packets per second).
latency	Estimated latency for this network interface card (msec).

Table 13 oclumon dumpnodeview FILESYSTEMS View Metric Descriptions

Metric	Description
total	Total amount of space (KB).
mount	Mount point.
type	File system type, whether local file system, NFS, or other.
used	Amount of used space (KB).
available	Amount of available space (KB).
used%	Percentage of used space (%)
ifree%	Percentage of free file nodes (%). Note: This metric is not available on Microsoft Windows systems.

Table 14 oclumon dumpnodeview PROTOCOL ERRORS View Metric Descriptions

Metric	Description
IPHdrErr	Number of input datagrams discarded due to errors in the IPv4 headers of the datagrams.

Table 14 (Cont.) oclumon dumpnodeview PROTOCOL ERRORS View Metric Descriptions

Metric	Description
IPAddrErr	Number of input datagrams discarded because the IPv4 address in their IPv4 header's destination field was not a valid address to be received at this entity.
IPUnkProto	Number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
IPReasFail	Number of failures detected by the IPv4 reassembly algorithm.
IPFragFail	Number of IPv4 discarded datagrams due to fragmentation failures.
TCPFailedConn	Number of times that TCP connections have made a direct transition to the CLOSED state from either the SYN-SENT state or the SYN-RCVD state, plus the number of times that TCP connections have made a direct transition to the LISTEN state from the SYN-RCVD state.
TCPEstRst	Number of times that TCP connections have made a direct transition to the CLOSED state from either the ESTABLISHED state or the CLOSE-WAIT state.
TCPRetraSeg	Total number of TCP segments retransmitted.
UDPUnkPort	Total number of received UDP datagrams for which there was no application at the destination port.
UDPRcvErr	Number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Table 15 oclumon dumpnodeview CPUS View Metric Descriptions

Metric	Description
<i>cpuid</i>	Virtual CPU.
<i>sys-usage</i>	CPU usage in system space.
<i>user-usage</i>	CPU usage in user space.
<i>nice</i>	Value of NIC for a specific CPU.
<i>usage</i>	CPU usage for a specific CPU.
<i>iowait</i>	CPU wait time for I/O operations.

Example 2 dumpnodeview -n

The following example dumps node views from `node1`, `node2`, and `node3` collected over the last 12 hours:

```
$ oclumon dumpnodeview -n node1 node2 node3 -last "12:00:00"
```

The following example displays node views from all nodes collected over the last 15 minutes at a 30-second interval:

```
$ oclumon dumpnodeview -allnodes -last "00:15:00" -i 30
```

Example 3 dumpnodeview -format csv

The following example shows how to use the option `-format csv` to output content in comma-separated values file format:

```
# oclumon dumpnodeview -format csv
```

```
dumpnodeview: Node name not given. Querying for the local host
```

```
-----  
Node: node1 Clock: '2016-09-02 11.18.00-0700' SerialNo:310668  
-----
```

```
SYSTEM:
```

```
"#pcpus", "#cores", "#vcpus", "cpuht", "chipname", "cpuusage[%]", "cpusys[%]", "cpuuser[%]",  
"cpunice[%]", "cpuiowait[%]", "cpusteal[%]", "cpuq", "physmemfree[KB]", "physmemtotal[KB]"  
,  
"mcache[KB]", "swapfree[KB]", "swaptotal[KB]", "hugepagetotal", "hugepagefree", "hugepages  
ize",  
"ior[KB/S]", "iow[KB/S]", "ios[#/S]", "swpin[KB/S]", "swpout[KB/S]", "pgin[#/S]", "pgout[/  
S]",  
"netr[KB/S]", "netw[KB/  
S]", "#procs", "#procsoncpu", "#procs_blocked", "#rtprocs", "#rtprocsoncpu",  
"#fds", "#sysfdlimit", "#disks", "#nics", "loadavg1", "loadavg5", "loadavg15", "#nicErrors"  
2,12,24,Y,"Intel(R) Xeon(R) CPU X5670 @ 2.93GHz",  
68.66,5.40,63.26,0.00,0.00,0.00,0,820240,  
73959636,61520568,4191424,4194300,0,0,  
2048,143,525,64,0,0,0,279,600.888,437.070,951,24,0,58,N/A,  
33120,6815744,13,5,19.25,17.67,16.09,0
```

```
TOPCONSUMERS:
```

```
"topcpu", "topprivmem", "topshm", "topfd", "topthread"  
"java(25047) 225.44", "java(24667) 1008360", "ora_lms1_prod_1(28913) 4985464", "polkit-  
gnome-au(20730) 1038", "java(2734) 209"
```

Example 4 dumpnodeview -procag

The following example shows how to output node views, aggregated by category: DBBG (DB backgrounds), DBFG (DB foregrounds), CLUST (Cluster), and OTHER (other processes).

```
# oclumon dumpnodeview -procag
```

```
-----  
Node: node1 Clock: '2016-09-02 11.14.15-0700' SerialNo:310623  
-----
```

```
PROCESS AGGREGATE:
```

cpuusage[%]	privatemem[KB]	maxshmem[KB]	#threads	#fd	#processes
category	sid				
0.62	45791348	4985200	187	10250	183
DBBG prod_1					
0.52	29544192	3322648	191	10463	187
DBBG webdb_1					
17.81	8451288	967924	22	511	22

DBFG	webdb_1					
	75.94	34930368	1644492	64	1067	64
DBFG	prod_1					
	3.42	3139208	120256	480	3556	25
CLUST						
	1.66	1989424	16568	1110	4040	471
OTHER						

Example 5 Node View Output

```
-----
Node: rwsak10 Clock: '2016-05-08 02.11.25-0800' SerialNo:155631
-----
```

SYSTEM:

```
#pcpus: 2 #vcpus: 24 cpuct: Y chipname: Intel(R) cpu: 1.23 cpug: 0
physmemfree: 8889492 physmemtotal: 74369536 mcache: 55081824 swapfree: 18480404
swaptotal: 18480408 hugepagetotal: 0 hugepagefree: 0 hugepagesize: 2048 ior: 132
iow: 236 ios: 23 swpin: 0 swpout: 0 pgin: 131 pgout: 235 netr: 72.404
netw: 97.511 procs: 969 procsconcpu: 6 rtprocs: 62 rtprocsconcpu N/A #fds: 32640
#sysfdlimit: 6815744 #disks: 9 #nics: 5 nicErrors: 0
```

TOP CONSUMERS:

```
topcpu: 'osysmond.bin(30981) 2.40' topprivmem: 'oraagent.bin(14599) 682496'
topshm: 'ora_dbw2_oss_3(7049) 2156136' topfd: 'ocssd.bin(29986) 274'
topthread: 'java(32255) 53'
```

CPUS:

```
cpu18: sys-2.93 user-2.15 nice-0.0 usage-5.8 iowait-0.0 steal-0.0
.
.
.
```

PROCESSES:

```
name: 'osysmond.bin' pid: 30891 #procfdlimit: 65536 cpuusage: 2.40 privmem: 35808
shm: 81964 #fd: 119 #threads: 13 priority: -100 nice: 0 state: S
.
.
.
```

DEVICES:

```
sdi ior: 0.000 iow: 0.000 ios: 0 qlen: 0 wait: 0 type: SYS
sdal ior: 0.000 iow: 61.495 ios: 629 qlen: 0 wait: 0 type: SYS
.
.
.
```

NICS:

```
lo netr: 39.935 netwr: 39.935 neteff: 79.869 nicerrors: 0 pktsin: 25
pktsout: 25 errsin: 0 errsout: 0 indiscarded: 0 outdiscarded: 0
inunicast: 25 innonunicast: 0 type: PUBLIC
eth0 netr: 1.412 netwr: 0.527 neteff: 1.939 nicerrors: 0 pktsin: 15
pktsout: 4 errsin: 0 errsout: 0 indiscarded: 0 outdiscarded: 0
inunicast: 15 innonunicast: 0 type: PUBLIC latency: <1
```

FILESYSTEMS:

```
mount: / type: rootfs total: 563657948 used: 78592012 available: 455971824
used%: 14 ifree%: 99 GRID_HOME
```

```
.
.
.
```

PROTOCOL ERRORS:

```
IPHdrErr: 0 IPAddrErr: 0 IPUnkProto: 0 IPReasFail: 0 IPFragFail: 0
TCPFailedConn: 5197 TCPEstRst: 717163 TCPRetraSeg: 592 UDPUnkPort: 103306
UDPRcvErr: 70
```

oclumon manage

Use the `oclumon manage` command to view and change configuration information from the system monitor service.

Syntax

```
oclumon manage -repos {{changeretentiontime time} | {changerepossize memory_size}} |
-get {key1 [key2 ...] | alllogger [-details] | mylogger [-details]}
```

Parameters

Table 16 oclumon manage Command Parameters

Parameter	Description
<code>-repos {{changeretentiontime <i>time</i>} {changerepossize <i>memory_size</i>}}</code>	<p>The <code>-repos</code> flag is required to specify the following Cluster Health Monitor repository-related options:</p> <ul style="list-style-type: none"><code>changeretentiontime <i>time</i></code>: Use this option to confirm that there is sufficient tablespace to hold the amount of Cluster Health Monitor data that can be accumulated in a specific amount of time. Note: This option <i>does not</i> change retention time.<code>changerepossize <i>memory_size</i></code>: Use this option to change the Cluster Health Monitor repository space limit to a specified number of MB Caution: If you decrease the space limit of the Cluster Health Monitor repository, then all data collected before the resizing operation is permanently deleted.

Table 16 (Cont.) oclumon manage Command Parameters

Parameter	Description
<code>-get key1 [key2 ...]</code>	<p>Use this option to obtain Cluster Health Monitor repository information using the following keywords:</p> <ul style="list-style-type: none"> <code>resize</code>: Size of the Cluster Health Monitor repository, in seconds <code>reppath</code>: Directory path to the Cluster Health Monitor repository <code>master</code>: Name of the master node <code>alllogger</code>: Special key to obtain a list of all nodes running Cluster Logger Service <code>mylogger</code>: Special key to obtain the node running the Cluster Logger Service which is serving the current node <ul style="list-style-type: none"> • <code>-details</code>: Use this option with <code>alllogger</code> and <code>mylogger</code> for listing nodes served by the Cluster Logger Service <p>You can specify any number of keywords in a space-delimited list following the <code>-get</code> flag.</p>
<code>-h</code>	Displays online help for the <code>oclumon manage</code> command

Usage Notes

- The local system monitor service must be running to change the retention time of the Cluster Health Monitor repository.
- The Cluster Logger Service must be running to change the retention time of the Cluster Health Monitor repository.

Example 6 oclumon manage

The following examples show commands and sample output:

```
$ oclumon manage -get MASTER
Master = node1

$ oclumon manage -get alllogger -details
Logger = node1
Nodes = node1,node2

$ oclumon manage -repos changeretentiontime 86400

$ oclumon manage -repos changereporesize 6000
```

oclumon version

Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.

Syntax

```
oclumon version
```

Example 7 oclumon version

This command produces output similar to the following:

```
Cluster Health Monitor (OS), Version 12.2.0.1.0 - Production Copyright 2007,  
2016 Oracle. All rights reserved.
```

Managing the Cluster Resource Activity Log

Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.

Failures can occur as a result of a problem with a resource, a hosting node, or the network.

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs. The cluster resource activity log also provides a unified view of the cause of resource failure.

Use the following commands to manage and view the contents of the cluster resource activity log:

- [crsctl query calog](#)
Query the cluster resource activity logs matching specific criteria.
- [crsctl get calog maxsize](#)
To store Oracle Clusterware-managed resource activity information, query the maximum space allotted to the cluster resource activity log.
- [crsctl get calog retentiontime](#)
Query the retention time of the cluster resource activity log.
- [crsctl set calog maxsize](#)
Configure the maximum amount of space allotted to store Oracle Clusterware-managed resource activity information.
- [crsctl set calog retentiontime](#)
Configure the retention time of the cluster resource activity log.

crsctl query calog

Query the cluster resource activity logs matching specific criteria.

Syntax

```
crsctl query calog [-aftertime "timestamp"] [-beforetime "timestamp"]  
  [-duration "time_interval" | -follow] [-filter "filter_expression"]  
  [-fullfmt | -xmlfmt]
```

Parameters

Table 17 crsctl query calog Command Parameters

Parameter	Description
<code>-aftertime "timestamp"</code>	<p>Displays the activities logged after a specific time.</p> <p>Specify the timestamp in the <code>YYYY-MM-DD HH24:MI:SS[.FF]</code> <code>[TZH:TZM]</code> or <code>YYYY-MM-DD</code> or <code>HH24:MI:SS[.FF]</code> <code>[TZH:TZM]</code> format.</p> <p><code>TZH</code> and <code>TZM</code> stands for time zone hour and minute, and <code>FF</code> stands for microseconds.</p> <p>If you specify <code>[TZH:TZM]</code>, then the <code>crsctl</code> command assumes UTC as time zone. If you do not specify <code>[TZH:TZM]</code>, then the <code>crsctl</code> command assumes the local time zone of the cluster node from where the <code>crsctl</code> command is run.</p> <p>Use this parameter with <code>-beforetime</code> to query the activities logged at a specific time interval.</p>
<code>-beforetime "timestamp"</code>	<p>Displays the activities logged before a specific time.</p> <p>Specify the timestamp in the <code>YYYY-MM-DD HH24:MI:SS[.FF]</code> <code>[TZH:TZM]</code> or <code>YYYY-MM-DD</code> or <code>HH24:MI:SS[.FF]</code> <code>[TZH:TZM]</code> format.</p> <p><code>TZH</code> and <code>TZM</code> stands for time zone hour and minute, and <code>FF</code> stands for microseconds.</p> <p>If you specify <code>[TZH:TZM]</code>, then the <code>crsctl</code> command assumes UTC as time zone. If you do not specify <code>[TZH:TZM]</code>, then the <code>crsctl</code> command assumes the local time zone of the cluster node from where the <code>crsctl</code> command is run.</p> <p>Use this parameter with <code>-aftertime</code> to query the activities logged at a specific time interval.</p>
<code>-duration "time_interval" -follow</code>	<p>Use <code>-duration</code> to specify a time interval that you want to query when you use the <code>-aftertime</code> parameter.</p> <p>Specify the timestamp in the <code>DD HH:MM:SS</code> format.</p> <p>Use <code>-follow</code> to display a continuous stream of activities as they occur.</p>
<code>-filter "filter_expression"</code>	<p>Query any number of fields in the cluster resource activity log using the <code>-filter</code> parameter.</p> <p>To specify multiple filters, use a comma-delimited list of filter expressions surrounded by double quotation marks ("").</p>
<code>-fullfmt -xmlfmt</code>	<p>To display cluster resource activity log data, choose full or XML format.</p>

Cluster Resource Activity Log Fields

Query any number of fields in the cluster resource activity log using the `-filter` parameter.

Table 18 Cluster Resource Activity Log Fields

Field	Description	Use Case
timestamp	The time when the cluster resource activities were logged.	Use this filter to query all the activities logged at a specific time. This is an alternative to - aftertime, -beforetime, and - duration command parameters.
writer_process_id	The ID of the process that is writing to the cluster resource activity log.	Query only the activities spawned by a specific process.
writer_process_name	The name of the process that is writing to the cluster resource activity log.	When you query a specific process, CR\$CTL returns all the activities for a specific process.
writer_user	The name of the user who is writing to the cluster resource activity log.	Query all the activities written by a specific user.
writer_group	The name of the group to which a user belongs who is writing to the cluster resource activity log.	Query all the activities written by users belonging to a specific user group.
writer_hostname	The name of the host on which the cluster resource activity log is written.	Query all the activities written by a specific host.
writer_clustername	The name of the cluster on which the cluster resource activity log is written.	Query all the activities written by a specific cluster.
nls_product	The product of the NLS message, for example, CRS, ORA, or srvm.	Query all the activities that have a specific product name.
nls_facility	The facility of the NLS message, for example, CRS or PROC.	Query all the activities that have a specific facility name.
nls_id	The ID of the NLS message, for example 42008.	Query all the activities that have a specific message ID.
nls_field_count	The number of fields in the NLS message.	Query all the activities that correspond to NLS messages with more than, less than, or equal to nls_field_count command parameters.
nls_field1	The first field of the NLS message.	Query all the activities that match the first parameter of an NLS message.
nls_field1_type	The type of the first field in the NLS message.	Query all the activities that match a specific type of the first parameter of an NLS message.
nls_format	The format of the NLS message, for example, Resource '%S' has been modified.	Query all the activities that match a specific format of an NLS message.

Table 18 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
nls_message	The entire NLS message that was written to the cluster resource activity log, for example, Resource 'ora.cvu' has been modified.	Query all the activities that match a specific NLS message.
actid	The unique activity ID of every cluster activity log.	Query all the activities that match a specific ID. Also, specify only partial actid and list all activities where the actid is a subset of the activity ID.
is_planned	Confirms if the activity is planned or not. For example, if a user issues the command <code>crsctl stop crs</code> on a node, then the stack stops and resources bounce. Running the <code>crsctl stop crs</code> command generates activities and logged in the calog. Since this is a planned action, the <code>is_planned</code> field is set to true (1). Otherwise, the <code>is_planned</code> field is set to false (0).	Query all the planned or unplanned activities.
onbehalfof_user	The name of the user on behalf of whom the cluster activity log is written.	Query all the activities written on behalf of a specific user.
entity_isoraentity	Confirms if the entity for which the calog activities are being logged is an oracle entity or not. If a resource, such as <code>ora.***</code> , is started or stopped, for example, then all those activities are logged in the cluster resource activity log. Since <code>ora.***</code> is an Oracle entity, the <code>entity_isoraentity</code> field is set to true (1). Otherwise the <code>entity_isoraentity</code> field is set to false (0).	Query all the activities logged by Oracle or non-Oracle entities.
entity_type	The type of the entity, such as <code>server</code> , for which the cluster activity log is written.	Query all the activities that match a specific entity.
entity_name	The name of the entity, for example, <code>foo</code> for which the cluster activity log is written.	Query all the cluster activities that match a specific entity name.

Table 18 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
entity_hostname	The name of the host, for example, <i>node1</i> , associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific host name.
entity_clustername	The name of the cluster, for example, <i>cluster1</i> associated with the entity for which the cluster activity log is written.	Query all the cluster activities that match a specific cluster name.

Usage Notes

Combine simple filters into expressions called expression filters using Boolean operators.

Enclose timestamps and time intervals in double quotation marks (").

Enclose the filter expressions in double quotation marks (").

Enclose the values that contain parentheses or spaces in single quotation marks (').

If no matching records are found, then the Oracle Clusterware Control (CRSCTL) utility displays the following message:

```
CRS-40002: No activities match the query.
```

Examples

Examples of filters include:

- "writer_user==root": Limits the display to only root user.
- "customer_data=='GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~'" : Limits the display to customer_data that has the specified value
GEN_RESTART@SERVERNAME(node1)=StartCompleted~.

To query all the resource activities and display the output in full format:

```
$ crsctl query calog -fullfmt
```

```
----ACTIVITY START----
timestamp           : 2016-09-27 17:55:43.152000
writer_process_id   : 6538
writer_process_name : crsd.bin
writer_user         : root
writer_group        : root
writer_hostname     : node1
writer_clustername  : cluster1-mbl
customer_data       : CHECK_RESULTS=-408040060~
nls_product         : CRS
nls_facility        : CRS
nls_id              : 2938
nls_field_count     : 1
nls_field1          : ora.cvu
nls_field1_type     : 25
nls_field1_len      : 0
nls_format          : Resource '%s' has been modified.
```

```

nls_message      : Resource 'ora.cvu' has been modified.
actid            : 14732093665106538/1816699/1
is_planned       : 1
onbehalfof_user  : grid
onbehalfof_hostname : node1
entity_isoraentity : 1
entity_type      : resource
entity_name      : ora.cvu
entity_hostname  : node1
entity_clustertype : cluster1-mbl
-----ACTIVITY END-----

```

To query all the resource activities and display the output in XML format:

```
$ crsctl query calog -xmlfmt
```

```

<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-27 17:55:43.152000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustertype>cluster1-mbl</writer_clustertype>
    <customer_data>CHECK_RESULTS=-408040060~</customer_data>
    <nls_product>CRS</nls_product>
    <nls_facility>CRS</nls_facility>
    <nls_id>2938</nls_id>
    <nls_field_count>1</nls_field_count>
    <nls_field1>ora.cvu</nls_field1>
    <nls_field1_type>25</nls_field1_type>
    <nls_field1_len>0</nls_field1_len>
    <nls_format>Resource '%s' has been modified.</nls_format>
    <nls_message>Resource 'ora.cvu' has been modified.</nls_message>
    <actid>14732093665106538/1816699/1</actid>
    <is_planned>1</is_planned>
    <onbehalfof_user>grid</onbehalfof_user>
    <onbehalfof_hostname>node1</onbehalfof_hostname>
    <entity_isoraentity>1</entity_isoraentity>
    <entity_type>resource</entity_type>
    <entity_name>ora.cvu</entity_name>
    <entity_hostname>node1</entity_hostname>
    <entity_clustertype>cluster1-mbl</entity_clustertype>
  </activity>
</activities>

```

To query resource activities for a two-hour interval after a specific time and display the output in XML format:

```

$ crsctl query calog -aftertime "2016-09-28 17:55:43" -duration "0 02:00:00" -xmlfmt
<?xml version="1.0" encoding="UTF-8"?>
<activities>
  <activity>
    <timestamp>2016-09-28 17:55:45.992000</timestamp>
    <writer_process_id>6538</writer_process_id>
    <writer_process_name>crsd.bin</writer_process_name>
    <writer_user>root</writer_user>
    <writer_group>root</writer_group>
    <writer_hostname>node1</writer_hostname>
    <writer_clustertype>cluster1-mbl</writer_clustertype>

```

```

<customer_data>CHECK_RESULTS=1718139884~</customer_data>
<nls_product>CRS</nls_product>
<nls_facility>CRS</nls_facility>
<nls_id>2938</nls_id>
<nls_field_count>1</nls_field_count>
<nls_field1>ora.cvu</nls_field1>
<nls_field1_type>25</nls_field1_type>
<nls_field1_len>0</nls_field1_len>
<nls_format>Resource '%s' has been modified.</nls_format>
<nls_message>Resource 'ora.cvu' has been modified.</nls_message>
<actid>14732093665106538/1942009/1</actid>
<is_planned>1</is_planned>
<onbehalfof_user>grid</onbehalfof_user>
<onbehalfof_hostname>nodel</onbehalfof_hostname>
<entity_isoraentity>1</entity_isoraentity>
<entity_type>resource</entity_type>
<entity_name>ora.cvu</entity_name>
<entity_hostname>nodel</entity_hostname>
<entity_clustername>cluster1-mbl</entity_clustername>
</activity>
</activities>

```

To query resource activities at a specific time:

```
$ crsctl query calog -filter "timestamp=='2016-09-28 17:55:45.992000'"
```

```
2016-09-28 17:55:45.992000 : Resource 'ora.cvu' has been modified. :
14732093665106538/1942009/1 :
```

To query resource activities using filters `writer_user` and `customer_data`:

```
$ crsctl query calog -filter "writer_user==root AND customer_data==
'GEN_RESTART@SERVERNAME(nodel)=StartCompleted~'" -fullfmt
```

or

```
$ crsctl query calog -filter "(writer_user==root) AND (customer_data==
'GEN_RESTART@SERVERNAME(nodel)=StartCompleted~')" -fullfmt
```

```

----ACTIVITY START----
timestamp           : 2016-09-15 17:42:57.517000
writer_process_id   : 6538
writer_process_name : crsd.bin
writer_user         : root
writer_group        : root
writer_hostname     : nodel
writer_clustername  : cluster1-mbl
customer_data       : GEN_RESTART@SERVERNAME(rwsbi08)=StartCompleted~
nls_product         : CRS
nls_facility        : CRS
nls_id              : 2938
nls_field_count     : 1
nls_field1          : ora.testdb.db
nls_field1_type     : 25
nls_field1_len      : 0
nls_format          : Resource '%s' has been modified.
nls_message         : Resource 'ora.devdb.db' has been modified.
actid               : 14732093665106538/659678/1
is_planned          : 1
onbehalfof_user     : oracle
onbehalfof_hostname : nodel
entity_isoraentity  : 1

```

```
entity_type          : resource
entity_name          : ora.testdb.db
entity_hostname      : node1
entity_clustername   : cluster1-mbl
----ACTIVITY END----
```

To query all the calogs that were generated after UTC+08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08+08:00"
```

To query all the calogs that were generated after UTC-08:00 time "2016-11-15 22:53:08":

```
$ crsctl query calog -aftertime "2016-11-15 22:53:08-08:00"
```

To query all the calogs by specifying the timestamp with microseconds:

```
$ crsctl query calog -aftertime "2016-11-16 01:07:53.063000"
```

```
2016-11-16 01:07:53.558000 : Resource 'ora.cvu' has been modified. :
14792791129816600/2580/7 :
2016-11-16 01:07:53.562000 : Clean of 'ora.cvu' on 'rwsam02' succeeded :
14792791129816600/2580/8 :
```

crsctl get calog maxsize

To store Oracle Clusterware-managed resource activity information, query the maximum space allotted to the cluster resource activity log.

Syntax

```
crsctl get calog maxsize
```

Parameters

The `crsctl get calog maxsize` command has no parameters.

Example

The following example returns the maximum space allotted to the cluster resource activity log to store activities:

```
$ crsctl get calog maxsize
```

```
CRS-6760: The maximum size of the Oracle cluster activity log is 1024 MB.
```

crsctl get calog retentiontime

Query the retention time of the cluster resource activity log.

Syntax

```
crsctl get calog retentiontime
```

Parameters

The `crsctl get calog retentiontime` command has no parameters.

Examples

The following example returns the retention time of the cluster activity log, in number of hours:

```
$ crsctl get calog retentiontime
```

```
CRS-6781: The retention time of the cluster activity log is 73 hours.
```

crsctl set calog maxsize

Configure the maximum amount of space allotted to store Oracle Clusterware-managed resource activity information.

Syntax

```
crsctl set calog maxsize maximum_size
```

Usage Notes

Specify a value, in MB, for the maximum size of the storage space that you want to allot to the cluster resource activity log.

Note:

If you reduce the amount of storage space, then the contents of the storage are lost.

Example

The following example sets maximum amount of space, to store Oracle Clusterware-managed resource activity information, to 1024 MB:

```
$ crsctl set calog maxsize 1024
```

crsctl set calog retentiontime

Configure the retention time of the cluster resource activity log.

Syntax

```
crsctl set calog retentiontime hours
```

Parameters

The `crsctl set calog retentiontime` command takes a number of hours as a parameter.

Usage Notes

Specify a value, in hours, for the retention time of the cluster resource activity log.

Examples

The following example sets the retention time of the cluster resource activity log to 72 hours:

```
$ crsctl set calog retentiontime 72
```

chactl Command Reference

The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

- [chactl monitor](#)
Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle Real Application Clusters (Oracle RAC) database using the current set model.
- [chactl unmonitor](#)
Use the `chactl unmonitor` command to stop monitoring all the instances of a specific database.
- [chactl status](#)
Use the `chactl status` command to check monitoring status of the running targets.
- [chactl config](#)
Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.
- [chactl calibrate](#)
Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- [chactl query diagnosis](#)
Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.
- [chactl query model](#)
Use the `chactl query model` command to list all Oracle Cluster Health Advisor models or to view detailed information about a specific Oracle Cluster Health Advisor model.
- [chactl query repository](#)
Use the `chactl query repository` command to view the maximum retention time, number of targets, and the size of the Oracle Cluster Health Advisor repository.
- [chactl query calibration](#)
Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- [chactl remove model](#)
Use the `chactl remove model` command to delete an Oracle Cluster Health Advisor model along with the calibration data and metadata of the model from the Oracle Cluster Health Advisor repository.
- [chactl rename model](#)
Use the `chactl rename model` command to rename an Oracle Cluster Health Advisor model in the Oracle Cluster Health Advisor repository.

- [chactl export model](#)
Use the `chactl export model` command to export Oracle Cluster Health Advisor models.
- [chactl import model](#)
Use the `chactl import model` command to import Oracle Cluster Health Advisor models.
- [chactl set maxretention](#)
Use the `chactl set maxretention` command to set the maximum retention time for the diagnostic data.
- [chactl resize repository](#)
Use the `chactl resize repository` command to resize the tablespace of the Oracle Cluster Health Advisor repository based on the current retention time and the number of targets.

chactl monitor

Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle Real Application Clusters (Oracle RAC) database using the current set model.

Oracle Cluster Health Advisor monitors all instances of this database using the same model assigned to the database.

Oracle Cluster Health Advisor uses Oracle-supplied gold model when you start monitoring a target for the first time. Oracle Cluster Health Advisor stores monitoring status of the target in the internal store. Oracle Cluster Health Advisor starts monitoring any new database instance when Oracle Cluster Health Advisor detects or redetects the new instance.

Syntax

```
chactl monitor database -db db_unique_name [-model model_name [-force]][-help]
```

```
chactl monitor cluster [-model model_name [-force]]
```

Parameters

Table 19 chactl monitor Command Parameters

Parameter	Description
<code>db_unique_name</code>	Specify the name of the database.
<code>model_name</code>	Specify the name of the model.
<code>force</code>	Use the <code>-force</code> option to monitor with the specified model without stopping monitoring the target. Without the <code>-force</code> option, run <code>chactl unmonitor</code> first, and then <code>chactl monitor</code> with the model name.

Examples

- To monitor the *SalesDB* database using the *BlkFridayShopping* default model:

```
$ chactl monitor database -db SalesDB -model BlkFridayShopping
```
- To monitor the *InventoryDB* database using the *Nov2014* model:

```
$ chactl monitor database -db InventoryDB -model Nov2014
```

If you specify the *model_name*, then Oracle Cluster Health Advisor starts monitoring with the specified model and stores the model in the Oracle Cluster Health Advisor internal store.

If you use both the *-model* and *-force* options, then Oracle Cluster Health Advisor stops monitoring and restarts monitoring with the specified model.

- To monitor the *SalesDB* database using the *Dec2014* model:

```
$ chactl monitor database -db SalesDB -model Dec2014
```

- To monitor the *InventoryDB* database using the *Dec2014* model and the *-force* option:

```
$ chactl monitor database -db InventoryDB -model Dec2014 -force
```

Error Messages

Error: no CHA resource is running in the cluster.

Description: Returns when there is no hub or leaf node running the Oracle Cluster Health Advisor service.

Error: the database is not configured.

Description: Returns when the database is not found in either the Oracle Cluster Health Advisor configuration repository or as a CRS resource.

Error: input string "xc#? %" is invalid.

Description: Returns when the command-line cannot be parsed. Also displays the top-level help text.

Error: CHA is already monitoring target <dbname>.

Description: Returns when the database is already monitored.

chactl unmonitor

Use the `chactl unmonitor` command to stop monitoring all the instances of a specific database.

Syntax

```
chactl unmonitor database -db db_unique_name [-help]
```

Examples

To stop monitoring the *SalesDB* database:

```
$ chactl unmonitor database -db SalesDB  
Database SalesDB is not monitored
```

chactl status

Use the `chactl status` command to check monitoring status of the running targets.

If you do not specify any parameters, then the `chactl status` command returns the status of all running targets.

The monitoring status of an Oracle Cluster Health Advisor target can be either **Monitoring** *or* **Not Monitoring**. The `chactl status` command shows four types of results and depends on whether you specify a target and `-verbose` option.

The `-verbose` option of the command also displays the monitoring status of targets contained within the specified target and the names of executing models of each printed target. The `chactl status` command displays targets with positive monitoring status only. The `chactl status` command displays negative monitoring status only when the corresponding target is explicitly specified on the command-line.

Syntax

```
chactl status {cluster|database [-db db_unique_name]} [-verbose][-help]
```

Examples

- To display the list of cluster nodes and databases being monitored:

```
#chactl status
Monitoring nodes rac1Node1, rac1Node2
Monitoring databases SalesDB, HRdb
```

Note:

A database is displayed with **Monitoring** status, if Oracle Cluster Health Advisor is monitoring one or more of the instances of the database, even if some of the instances of the database are not running.

- To display the status of Oracle Cluster Health Advisor:

```
$ chactl status
Cluster Health Advisor service is offline.
```

No target or the `-verbose` option is specified on the command-line. Oracle Cluster Health Advisor is not running on any node of the cluster.

- To display various Oracle Cluster Health Advisor monitoring states for cluster nodes and databases:

```
$ chactl status database -db SalesDB
Monitoring database SalesDB
```

```
$ chactl status database -db bogusDB
Not Monitoring database bogusDB
```

```
$ chactl status cluster
Monitoring nodes rac1,rac2
Not Monitoring node rac3
```

or

```
$ chactl status cluster
Cluster Health Advisor is offline
```

- To display the detailed Oracle Cluster Health Advisor monitoring status for the entire cluster:

```
$ chactl status -verbose
Monitoring node(s) racNd1, racNd2, racNd3, racNd4 using model MidSparc

Monitoring database HRdb2, Instances HRdb2I1, HRdb2I2 in server pool SilverPool
using model M6
Monitoring database HRdb, Instances HRdbI4, HRdbI6 in server pool SilverPool
using model M23
Monitoring database testHR, Instances inst3 on node racN7 using model TestM13
Monitoring database testHR, Instances inst4 on node racN8 using model TestM14
```

When the target is not specified and the `-verbose` option is specified, the `chactl status` command displays the status of the database instances and names of the models.

chactl config

Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.

If the specified target is a multitenant container database (CDB) or a cluster, then the `chactl config` command also displays the configuration data status.

Syntax

```
chactl config {cluster|database -db db_unique_name}[-help]
```

Examples

To display the monitor configuration and the specified model of each target:

```
$ chactl config
Databases monitored: prodDB, hrDB

$ chactl config database -db prodDB
Monitor: Enabled
Model: GoldDB

$ chactl config cluster
Monitor: Enabled
Model: DEFAULT_CLUSTER
```

chactl calibrate

Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.

The user-generated models are effective for Oracle Real Application Clusters (Oracle RAC) monitored systems in your operating environment as the user-generated models use calibration data from the target. Oracle Cluster Health Advisor adds the user-generated model to the list of available models and stores the new model in the Oracle Cluster Health Advisor repository.

If a model with the same name exists, then overwrite the old model with the new one by using the `-force` option.

Key Performance and Workload Indicators

A set of metrics or Key Performance Indicators describe high-level constraints to the training data selected for calibration. This set consists of relevant metrics to describe performance goals and resource utilization bandwidth, for example, response times or CPU utilization.

The Key Performance Indicators are also operating system and database signals which are monitored, estimated, and associated with fault detection logic. Most of these Key Performance Indicators are also either predictors, that is, their state is correlated with the state of other signals, or predicted by other signals. The fact that the Key Performance Indicators correlate with other signals makes them useful as filters for the training or calibration data.

The Key Performance Indicators ranges are used in the `query calibrate` and `calibrate` commands to filter out data points.

The following Key Performance Indicators are supported for database:

- `CPUPERCENT` - CPU utilization - Percent
- `IOREAD` - Disk read - Mbyte/sec
- `DBTIMEPERCALL` - Database time per user call - usec/call
- `IOWRITE` - Disk write - Mbyte/sec
- `IOTROUGHPUT` - Disk throughput - IO/sec

The following Key Performance Indicators are supported for cluster:

- `CPUPERCENT` - CPU utilization - Percent
- `IOREAD` - Disk read - Mbyte/sec
- `IOWRITE` - Disk write - Mbyte/sec
- `IOTROUGHPUT` - Disk throughput - IO/sec

Syntax

```
chactl calibrate {cluster|database -db db_unique_name} -model model_name
[-force] [-timeranges 'start=time_stamp,end=time_stamp,...']
[-kpiiset 'name=kpi_name min=val max=val,...' ][-help]
```

Specify timestamp in the `YYYY-MM-DD HH24:MI:SS` format.

Examples

```
chactl calibrate database -db oracle -model weekday
-timeranges 'start=start=2016-09-09 16:00:00,end=2016-09-09 23:00:00'
```

```
chactl calibrate database -db oracle -model weekday
-timeranges 'start=start=2016-09-09 16:00:00,end=2016-09-09 23:00:00'
-kpiiset 'name=CPUPERCENT min=10 max=60'
```

Error Messages

Error: input string "`xc#? %`" is misconstructured

Description: Confirm if the given model name exists with warning: *model_name* already exists, please use [-force] message.

Error: *start_time* and/or *end_time* are misconstructured

Description: Input time specifiers are badly constructed.

Error: no sufficient calibration data exists for the specified period, please reselect another period

Description: Evaluator couldn't find enough calibration data.

chactl query diagnosis

Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.

Syntax

```
chactl query diagnosis [-cluster|-db db_unique_name] [-start time -end time] [-htmlfile file_name][-help]
```

Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

In the preceding syntax, you must consider the following points:

- If you do not provide any options, then the `chactl query diagnosis` command returns the current state of all monitored nodes and databases. The `chactl query diagnosis` command reports general state of the targets, for example, **ABNORMAL** by showing their diagnostic identifier, for example, Storage Bandwidth Saturation. This is a quick way to check for any **ABNORMAL** state in a database or cluster.
- If you provide a time option after the target name, then the `chactl query diagnosis` command returns the state of the specified target restricted to the conditions in the time interval specified. The compressed time series lists the identifiers of the causes for distinct incidents which occurred in the time interval, its start and end time.
- If an incident and cause recur in a specific time interval, then the problem is reported only once. The start time is the start time of the first occurrence of the incident and the end time is the end time of the last occurrence of the incident in the particular time interval.
- If you specify the `-db` option without a database name, then the `chactl query diagnosis` command displays diagnostic information for all databases. However, if a database name is specified, then the `chactl query diagnosis` command displays diagnostic information for all instances of the database that are being monitored.
- If you specify the `-cluster` option without a host name, then the `chactl query diagnosis` command displays diagnostic information for all hosts in that cluster.
- If you do not specify a time interval, then the `chactl query diagnosis` command displays only the current issues for all or the specified targets. The `chactl query diagnosis` command does not display the frequency statistics explicitly. However, you can count the number of normal and abnormal events that occurred in a target in the last 24 hours.

- If no incidents have occurred during the specified time interval, then the `chactl query diagnosis` command returns a text message, for example, Database/host is operating NORMALLY, or no incidents were found.
- If the state of a target is **NORMAL**, the command does not report it. The `chactl query diagnosis` command reports only the targets with **ABNORMAL** state for the specified time interval.

Output parameters:

- Incident start Time
- Incident end time (only for the default database and/or host, non-verbose output)
- Target (for example, database, host)
- Problem
 - Description: Detailed description of the problem
 - Cause: Root cause of the problem and contributing factors
- Action: an action that corrects the abnormal state covered in the diagnosis

Reporting Format: The diagnostic information is displayed in a time compressed or time series order, grouped by components.

Examples

To display diagnostic information of a database for a specific time interval:

```
$ chactl query diagnosis -db oltpacdb -start "2016-02-01 02:52:50.0" -end
"2016-02-01 03:19:15.0"
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_1) [detected]
2016-02-01 01:47:10.0 Database oltpacdb DB Control File IO Performance
(oltpacdb_2) [detected]
2016-02-01 02:52:15.0 Database oltpacdb DB CPU Utilization (oltpacdb_2) [detected]
2016-02-01 02:52:50.0 Database oltpacdb DB CPU Utilization (oltpacdb_1) [detected]
2016-02-01 02:59:35.0 Database oltpacdb DB Log File Switch (oltpacdb_1) [detected]
2016-02-01 02:59:45.0 Database oltpacdb DB Log File Switch (oltpacdb_2) [detected]
```

Problem: DB Control File IO Performance

Description: CHA has detected that reads or writes to the control files are slower than expected.

Cause: The Cluster Health Advisor (CHA) detected that reads or writes to the control files were slow

because of an increase in disk IO.

The slow control file reads and writes may have an impact on checkpoint and Log Writer (LGWR) performance.

Action: Separate the control files from other database files and move them to faster disks or Solid State Devices.

Problem: DB CPU Utilization

Description: CHA detected larger than expected CPU utilization for this database.

Cause: The Cluster Health Advisor (CHA) detected an increase in database CPU utilization

because of an increase in the database workload.

Action: Identify the CPU intensive queries by using the Automatic Diagnostic and Defect Manager (ADDM)

and follow the recommendations given there. Limit the number of CPU intensive queries or relocate sessions to less busymachines. Add CPUs if the CPU capacity is insufficient to support the load

without a performance degradation or effects on other databases.

Problem: DB Log File Switch
Description: CHA detected that database sessions are waiting longer than expected for log switch completions.
Cause: The Cluster Health Advisor (CHA) detected high contention during log switches because the redo log files were small and the redo logs switched frequently.
Action: Increase the size of the redo logs.

Error Message

Message: *Target* is operating normally

Description: No incidents are found on the target.

Message: No data was found for active *Target*

Description: No data was found, but the target was operating or active at the time of the query.

Message: Target is not active or was not being monitored.

Description: No data was found because the target was not monitored at the time of the query.

chactl query model

Use the `chactl query model` command to list all Oracle Cluster Health Advisor models or to view detailed information about a specific Oracle Cluster Health Advisor model.

Syntax

```
chactl query model [-name model_name [-verbose]][-help]
```

Examples

- To list all base Oracle Cluster Health Advisor models:

```
$ chactl query model  
Models: MOD1, MOD2, MOD3, MOD4, MOD5, MOD6, MOD7
```

```
$ chactl query model -name weekday  
Model: weekday  
Target Type: DATABASE  
Version: 12.2.0.1_0  
OS Calibrated on: Linux amd64  
Calibration Target Name: prod  
Calibration Date: 2016-09-10 12:59:49  
Calibration Time Ranges: start=2016-09-09 16:00:00,end=2016-09-09 23:00:00  
Calibration KPIs: not specified
```

- To view detailed information, including calibration metadata, about the specific Oracle Cluster Health Advisor model:

```
$ chactl query model -name MOD5 -verbose  
Model: MOD5  
CREATION_DATE:          Jan 10,2016 10:10  
VALIDATION_STATUS:     Validated  
DATA_FROM_TARGET :     inst72, inst75  
USED_IN_TARGET :       inst76, inst75, prodDB, evalDB-evalSP  
CAL_DATA_FROM_DATE:    Jan 05,2016 10:00  
CAL_DATA_TO_DATE:      Jan 07,2016 13:00
```

```
CAL_DATA_FROM_TARGETS  inst73, inst75
...
```

chactl query repository

Use the `chactl query repository` command to view the maximum retention time, number of targets, and the size of the Oracle Cluster Health Advisor repository.

Syntax

```
chactl query repository [-help]
```

Examples

To view information about the Oracle Cluster Health Advisor repository:

```
$ chactl query repository
specified max retention time(hrs) : 72
available retention time(hrs)     : 212
available number of entities      : 2
allocated number of entities      : 0
total repository size(gb)        : 2.00
allocated repository size(gb)    : 0.07
```

chactl query calibration

Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.

Syntax

```
chactl query calibration {-cluster|-db db_unique_name} [-timeranges
'start=time_stamp,end=time_stamp,...'] [-kpiiset 'name=kpi_name min=val
max=val,...' ] [-interval val][-help]
```

Specify the interval in hours.

Specify date and time in the YYYY-MM-DD HH24:MI:SS format.

Note:

If you do not specify a time interval, then the `chactl query calibration` command displays all the calibration data collected for a specific target.

The following Key Performance Indicators are supported for database:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- DBTIMEPERCALL - Database time per user call - usec/call
- IOWRITE - Disk write - Mbyte/sec
- IOTHROUGHPUT - Disk throughput - IO/sec

The following Key Performance Indicators are supported for cluster:

- CPUPERCENT - CPU utilization - Percent
- IOREAD - Disk read - Mbyte/sec
- IOWRITE - Disk write - Mbyte/sec
- IOTHROUGHPUT - Disk throughput - IO/sec

Examples

To view detailed information about the calibration data of the specified target:

```
$ chactl query calibration -db oltpacdb -timeranges
'start=2016-07-26 01:00:00,end=2016-07-26 02:00:00,start=2016-07-26
03:00:00,end=2016-07-26 04:00:00'
-kpiset 'name=CPUPERCENT min=20 max=40, name=IOTHROUGHPUT min=500 max=9000' -
interval 2
```

```
Database name : oltpacdb
Start time : 2016-07-26 01:03:10
End time : 2016-07-26 01:57:25
Total Samples : 120
Percentage of filtered data : 8.32%
The number of data samples may not be sufficient for calibration.
```

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
4.96	0.20	8.98	0.06	25.68
<25	<50	<75	<100	>=100
97.50%	2.50%	0.00%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
27.73	9.72	31.75	4.16	109.39
<50	<100	<150	<200	>=200
73.33%	22.50%	4.17%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
2407.50	1500.00	1978.55	700.00	7800.00
<5000	<10000	<15000	<20000	>=20000
83.33%	16.67%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
21.99	21.75	1.36	20.00	26.80
<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX
267.39	264.87	32.05	205.80	484.57

<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000	>=70000000
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Database name : oltpacdb
 Start time : 2016-07-26 03:00:00
 End time : 2016-07-26 03:53:30
 Total Samples : 342
 Percentage of filtered data : 23.72%
 The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
12.18	0.28	16.07	0.05	60.98
<25	<50	<75	<100	>=100
64.33%	34.50%	1.17%	0.00%	0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
57.57	51.14	34.12	16.10	135.29
<50	<100	<150	<200	>=200
49.12%	38.30%	12.57%	0.00%	0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN	MEDIAN	STDDEV	MIN	MAX
5048.83	4300.00	1730.17	2700.00	9000.00
<5000	<10000	<15000	<20000	>=20000
63.74%	36.26%	0.00%	0.00%	0.00%

4) CPU utilization (total) (%)

MEAN	MEDIAN	STDDEV	MIN	MAX
23.10	22.80	1.88	20.00	31.40
<20	<40	<60	<80	>=80
0.00%	100.00%	0.00%	0.00%	0.00%

5) Database time per user call (usec/call)

MEAN	MEDIAN	STDDEV	MIN	MAX			
744.39	256.47	2892.71	211.45	45438.35			
<10000000	<20000000	<30000000	<40000000	<50000000	<60000000	<70000000	>=70000000
100.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

chactl remove model

Use the `chactl remove model` command to delete an Oracle Cluster Health Advisor model along with the calibration data and metadata of the model from the Oracle Cluster Health Advisor repository.



Note:

If the model is being used to monitor the targets, then the `chactl remove model` command cannot delete any model.

Syntax

```
chactl remove model -name model_name [-help]
```

Error Message

Error: *model_name* does not exist

Description: The specified Oracle Cluster Health Advisor model does not exist in the Oracle Cluster Health Advisor repository.

chactl rename model

Use the `chactl rename model` command to rename an Oracle Cluster Health Advisor model in the Oracle Cluster Health Advisor repository.

Assign a descriptive and unique name to the model. Oracle Cluster Health Advisor preserves all the links related to the renamed model.

Syntax

```
chactl rename model -from model_name -to model_name [-help]
```

Error Messages

Error: *model_name* does not exist

Description: The specified model name does not exist in the Oracle Cluster Health Advisor repository.

Error: *dest_name* already exist

Description: The specified model name already exists in the Oracle Cluster Health Advisor repository.

chactl export model

Use the `chactl export model` command to export Oracle Cluster Health Advisor models.

Syntax

```
chactl export model -name model_name -file output_file [-help]
```

Example

```
$ chactl export model -name weekday -file /tmp//weekday.mod
```

chactl import model

Use the `chactl import model` command to import Oracle Cluster Health Advisor models.

Syntax

```
chactl import model -name model_name -file model_file [-force] [-help]
```

While importing, if there is an existing model with the same name as the model being imported, then use the `-force` option to overwrite.

Example 8 Example

```
$ chactl import model -name weekday -file /tmp//weekday.mod
```

chactl set maxretention

Use the `chactl set maxretention` command to set the maximum retention time for the diagnostic data.

The default and minimum retention time is 72 hours. If the Oracle Cluster Health Advisor repository does not have enough space, then the retention time is decreased for all the targets.

Note:

Oracle Cluster Health Advisor stops monitoring if the retention time is less than 24 hours.

Syntax

```
chactl set maxretention -time retention_time [-help]
```

Specify the retention time in hours.

Examples

To set the maximum retention time to 80 hours:

```
$ chactl set maxretention -time 80  
max retention successfully set to 80 hours
```

Error Message

Error: Specified time is smaller than the allowed minimum

Description: This message is returned if the input value for maximum retention time is smaller than the minimum value.

chactl resize repository

Use the `chactl resize repository` command to resize the tablespace of the Oracle Cluster Health Advisor repository based on the current retention time and the number of targets.

Note:

The `chactl resize repository` command fails if your system does not have enough free disk space or if the tablespace contains data beyond requested resize value.

Syntax

```
chactl resize repository -entities total number of hosts and database instances [-force | -eval] [-help]
```

Examples

To set the number of targets in the tablespace to 32:

```
chactl resize repository -entities 32  
repository successfully resized for 32 targets
```

Oracle Trace File Analyzer Command-Line and Shell Options

The Trace File Analyzer control utility, TFACTL, is the command-line interface for Oracle Trace File Analyzer.

TFACTL provides a command-line and shell interface to Oracle Trace File Analyzer commands for:

- Administration
- Summary and analysis
- Diagnostic collection

The `tfactl` commands that you can run depends on your access level.

- You need `root` access or `sudo` access to `tfactl` to run administration commands.
- Run a subset of commands as:
 - An Oracle Database home owner or Oracle Grid Infrastructure home owner
 - A member of `OS DBA` or `ASM` groups

You gain access to summary, analysis, and diagnostic collection functionality by running the commands as an Oracle Database home owner or Oracle Grid Infrastructure home owner.

To grant other users access to `tfactl`:

```
tfactl access
```

To use `tfactl` as a command-line tool:

```
tfactl command [options]
```

To use `tfactl` as a shell interface:

```
tfactl
```

Once the shell starts enter commands as needed.

```
$ tfactl
```

```
tfactl>
```

Append the `-help` option to any of the `tfactl` commands to obtain command-specific help.

```
$ tfactl command -help
```

- [Running Administration Commands](#)
You need `root` access to `tfactl`, or `sudo` access to run all administration commands.
- [Running Summary and Analysis Commands](#)
Use these commands to view the summary of deployment and status of Oracle Trace File Analyzer, and changes and events detected by Oracle Trace File Analyzer.
- [Running Diagnostic Collection Commands](#)
Run the diagnostic collection commands to collect diagnostic data.

Running Administration Commands

You need `root` access to `tfactl`, or `sudo` access to run all administration commands.

Table 20 Basic TFACTL commands

Command	Description
<code>tfactl start</code>	Starts the Oracle Trace File Analyzer daemon on the local node.
<code>tfactl stop</code>	Stops the Oracle Trace File Analyzer daemon on the local node.
<code>tfactl enable</code>	Enables automatic restart of the Oracle Trace File Analyzer daemon after a failure or system reboot.
<code>tfactl disable</code>	Stops any running Oracle Trace File Analyzer daemon and disables automatic restart.
<code>tfactl uninstall</code>	Removes Oracle Trace File Analyzer from the local node.
<code>tfactl syncnodes</code>	Generates and copies Oracle Trace File Analyzer certificates from one Oracle Trace File Analyzer node to other nodes.
<code>tfactl restrictprotocol</code>	Restricts the use of certain protocols.

Table 20 (Cont.) Basic TFACTL commands

Command	Description
<code>tfactl status</code>	Checks the status of an Oracle Trace File Analyzer process. The output is same as <code>tfactl print status</code> .

- [tfactl diagnosetfa](#)
Use the `tfactl diagnosetfa` command to collect Oracle Trace File Analyzer diagnostic data from the local node to identify issues with Oracle Trace File Analyzer.
- [tfactl host](#)
Use the `tfactl host` command to add hosts to, or remove hosts from the Oracle Trace File Analyzer configuration.
- [tfactl set](#)
Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.
- [tfactl access](#)
Use the `tfactl access` command to allow non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections.

tfactl diagnosetfa

Use the `tfactl diagnosetfa` command to collect Oracle Trace File Analyzer diagnostic data from the local node to identify issues with Oracle Trace File Analyzer.

Syntax

```
tfactl diagnosetfa [-repo repository] [-tag tag_name] [-local]
```

Parameters

Table 21 tfactl diagnosetfa Command Parameters

Parameter	Description
<code>-repo repository</code>	Specify the repository directory for Oracle Trace File Analyzer diagnostic collections.
<code>-tag tag_name</code>	Oracle Trace File Analyzer collects the files into <code>tag_name</code> directory.
<code>-local</code>	Runs Oracle Trace File Analyzer diagnostics only on the local node.

tfactl host

Use the `tfactl host` command to add hosts to, or remove hosts from the Oracle Trace File Analyzer configuration.

Syntax

```
tfactl host [add host_name | remove host_name]
```

Specify a host name to add or remove, as in the following example:

```
$ tfactl host add myhost.example.com
```

Usage Notes

View the current list of hosts in the Oracle Trace File Analyzer configuration using the `tfactl print hosts` command. The `tfactl print hosts` command lists the hosts that are part of the Oracle Trace File Analyzer cluster:

```
$ tfactl print hosts
Host Name : node1
Host Name : node2
```

When you add a new host, Oracle Trace File Analyzer contacts the Oracle Trace File Analyzer instance on the other host. Oracle Trace File Analyzer authenticates the new host using certificates and both the Oracle Trace File Analyzer instances synchronize their respective hosts lists. Oracle Trace File Analyzer does not add the new host until the certificates are synchronized.

After you successfully add a host, all the cluster-wide commands are activated on all nodes registered in the Berkeley database.

tfactl set

Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

Syntax

```
tfactl set [autodiagcollect=ON | OFF] [cookie=UID] [autopurge=ON | OFF]
[minagetopurge=n]
[trimfiles=ON | OFF] [tracelevel=COLLECT | SCAN | INVENTORY | OTHER:1 | 2 | 3 | 4]
[manageLogsAutoPurge=ON | OFF] [manageLogsAutoPurgePolicyAge=nd|h]
[manageLogsAutoPurgeInterval=minutes] [diskUsageMon=ON|OFF]
[diskUsageMonInterval=minutes] [resizeMB=number]
[repositorydir=directory] [logsize=n [-local]] [logcount=n
[-local]] [-c]
```

Parameters

Table 22 tfactl set Command Parameters

Parameter	Description
autodiagcollect=ON OFF	When set to OFF (default) automatic diagnostic collection is disabled. If set to ON, then Oracle Trace File Analyzer automatically collects diagnostics when certain patterns occur while Oracle Trace File Analyzer scans the alert logs. To set automatic collection for all nodes of the Oracle Trace File Analyzer cluster, you must specify the <code>-c</code> parameter.
autopurge	When set to ON, enables automatic purging of collections when Oracle Trace File Analyzer observes less space in the repository (default is ON).
minagetopurge=n	Set the minimum age, in hours, for a collection before Oracle Trace File Analyzer considers it for purging (default is 12 hours).

Table 22 (Cont.) tfactl set Command Parameters

Parameter	Description
<code>trimfiles=ON OFF</code>	When set to ON, Oracle Trace File Analyzer trims the files to have only the relevant data when diagnostic collection is done as part of a scan. Note: When using <code>tfactl diagcollect</code> , you determine the time range for trimming with the parameters you specify. Oracle recommends that you <i>not</i> set this parameter to OFF, because untrimmed data can consume much space.
<code>tracelevel=COLLECT SCAN INVENTORY OTHER: 1 2 3 4</code>	You can set trace levels for certain operations, including INVENTORY: <i>n</i> , SCAN: <i>n</i> , COLLECT: <i>n</i> , OTHER: <i>n</i> . In this syntax, <i>n</i> is a number from 1 to 4 and OTHER includes all messages not relevant to the first three components. Note: Do not change the tracing level unless you are directed to do so by My Oracle Support.
<code>diskUsageMon=ON OFF</code>	Turns ON (default) or OFF monitoring disk usage and recording snapshots. Oracle Trace File Analyzer stores the snapshots under <code>tfa/repository/suptools/node/managerlogs/usage_snapshot/</code> .
<code>diskUsageMonInterval=m inutes</code>	Specify the time interval between snapshots (60 minutes by default).
<code>manageLogsAutoPurge=ON OFF</code>	Turns automatic purging on or off (ON by default in DSC and OFF by default elsewhere).
<code>manageLogsAutoPurgePol icyAge=nd h</code>	Age of logs to be purged (30 days by default).
<code>manageLogsAutoPurgeInt erval=minutes</code>	Specify the purge frequency (default is 60 minutes).
<code>resizeMB=number</code>	Sets the maximum size, in MB, of the collection repository.
<code>repositorydir=director y</code>	Specify the collection repository directory.
<code>logsize=n [-local]</code>	Sets the maximum size, in MB, of each log before Oracle Trace File Analyzer rotates to a new log (default is 50 MB). Use the <code>-local</code> parameter to apply the change only to the local node.
<code>logcount=n [-local]</code>	Sets the maximum number of logs of specified size that Oracle Trace File Analyzer retains (default is 10). Use the <code>-local</code> parameter to apply the change only to the local node.
<code>-c</code>	Propagates the settings to all nodes in the Oracle Trace File Analyzer configuration.

Example

The following example enables automatic diagnostic collection, sets the trace level, and sets a maximum limit for the collection repository:

```
$ tfactl set autodiagcollect=ON resizeMB=20480
```

tfactl access

Use the `tfactl access` command to allow non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections.

Non-root users can run a subset of `tfactl` commands. Running a subset of commands enables non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections. However, `root` access is still required to install and administer Oracle Trace File Analyzer. Control non-root users and groups using the `tfactl access` command. Add or remove non-root users and groups depending upon your business requirements.

Note:

By default, all Oracle home owners, OS DBA groups, and ASM groups are added to the Oracle Trace File Analyzer Access Manager list while installing or upgrading Oracle Trace File Analyzer.

Syntax

```
tfactl access [ lsusers | add -user user_name [ -group group_name ]
[ -local ] | remove -user user_name [ -group group_name ]
[ -all ] [ -local ] | block -user user_name [ -local ] | unblock -user user_name
[-local] | enable [ -local ] | disable [ -local ] | reset [ -local ] | removeall [ -
local ]
```

Parameters

Table 23 tfactl access Command Parameters

Parameter	Description
lsusers	Lists all the Oracle Trace File Analyzer users and groups.
enable	Enables Oracle Trace File Analyzer access for non-root users. Use the <code>-local</code> flag to change settings only on the local node.
disable	Disables Oracle Trace File Analyzer access for non-root users. However, the list of users who were granted access to Oracle Trace File Analyzer is stored, if the access to non-root users is enabled later. Use the <code>-local</code> flag to change settings only on the local node.
add	Adds a user or a group to the Oracle Trace File Analyzer access list.
remove	Removes a user or a group from the Oracle Trace File Analyzer access list.
block	Blocks Oracle Trace File Analyzer access for non-root user. Use this command to block a specific user even though the user is a member of a group that is granted access to Oracle Trace File Analyzer.

Table 23 (Cont.) tfactl access Command Parameters

Parameter	Description
unlock	Enables Oracle Trace File Analyzer access for non-root users who were blocked earlier. Use this command to unblock a user that was blocked earlier by running the command <code>tfactl access block</code> .
reset	Resets to the default access list that includes all Oracle Home owners and DBA groups.
removeall	Removes all Oracle Trace File Analyzer users and groups. Remove all users from the Oracle Trace File Analyzer access list including the default users and groups.

Examples

To add a user, for example, *abc* to the Oracle Trace File Analyzer access list and enable access to Oracle Trace File Analyzer across cluster.

```
/u01/app/tfa/bin/tfactl access add -user abc
```

To add all members of a group, for example, *xyz* to the Oracle Trace File Analyzer access list and enable access to Oracle Trace File Analyzer on the localhost.

```
/u01/app/tfa/bin/tfactl access add -group xyz -local
```

To remove a user, for example, *abc* from the Oracle Trace File Analyzer access list.

```
/u01/app/tfa/bin/tfactl access remove -user abc
```

To block a user, for example, *xyz* from accessing Oracle Trace File Analyzer.

```
/u01/app/tfa/bin/tfactl access block -user xyz
```

To remove all Oracle Trace File Analyzer users and groups.

```
/u01/app/tfa/bin/tfactl access removeall
```

Running Summary and Analysis Commands

Use these commands to view the summary of deployment and status of Oracle Trace File Analyzer, and changes and events detected by Oracle Trace File Analyzer.

- [tfactl summary](#)
Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.
- [tfactl changes](#)
Use the `tfactl changes` command to view the changes detected by Oracle Trace File Analyzer.
- [tfactl events](#)
Use the `tfactl events` command to view the events detected by Oracle Trace File Analyzer.

- [tfactl analyze](#)
Use the `tfactl analyze` command to obtain analysis of your system by parsing the database, Oracle ASM, and Oracle Grid Infrastructure alert logs, system message logs, OSWatcher Top, and OSWatcher Slabinfo files.
- [tfactl run](#)
Use the `tfactl run` command to run the requested action (can be inventory or scan or any support tool).
- [tfactl toolstatus](#)
Use the `tfactl toolstatus` command to view the status of Oracle Trace File Analyzer Support Tools across all nodes.

tfactl summary

Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.

Syntax

```
tfactl summary
```

Example

```
$ tfactl summary
Output from host : myserver69
```

```
-----
=====
Nodes
=====
myserver69
myserver70
myserver71

=====
Homes
=====
.-----
-----
| Home                                     | Type | Version   |
Database           | Instance           | Patches |
+-----+-----+-----+
| /scratch/app/11.2.0.4/grid                | GI   | 11.2.0.4.0 |
|                                           |     |           |
| /scratch/app/oradb/product/11.2.0/dbhome_11204 | DB   | 11.2.0.4.0 |
apxcmupg,rdb11204 | apxcmupg_1,rdb112041 |         |
'-----+-----+-----'
+-----+-----+-----'
```

```
Output from host : myserver70
```

```
-----
=====
Homes
=====
.-----
-----
| Home                                     | Type | Version   |
```

```

Database          | Instance | Patches |
+-----+-----+-----+
| /scratch/app/11.2.0.4/grid          | GI | 11.2.0.4.0
|                                     |   |
| /scratch/app/oradb/product/11.2.0/dbhome_11204 | DB | 11.2.0.4.0 |
apxcmupg,rdb11204 | rdb112042 |
'-----+-----+-----'
Output from host : myserver71
-----

```

```

=====
Homes
=====

```

```

-----
| Home                                     | Type | Version |
Database          | Instance | Patches |
+-----+-----+-----+
| /scratch/app/11.2.0.4/grid          | GI | 11.2.0.4.0
|                                     |   |
| /scratch/app/oradb/product/11.2.0/dbhome_11204 | DB | 11.2.0.4.0 |
apxcmupg,rdb11204 | rdb112043 |
'-----+-----+-----'
-----

```

tfactl changes

Use the `tfactl changes` command to view the changes detected by Oracle Trace File Analyzer.

Syntax

```
tfactl changes
```

Example

```
$ tfactl changes
```

```
Output from host : myserver69
-----
```

```
Output from host : myserver70
-----
```

```

Jul/26/2016 10:20:35 : Parameter 'sunrpc.transports' value changed : tcp 1048576 =>
udp 32768
Jul/26/2016 10:20:35 : Parameter 'sunrpc.transports' value changed : tcp 1048576 =>
tcp-bc 1048576

```

```
Output from host : myserver71
-----
```

```

Jul/26/2016 10:21:06 : Parameter 'sunrpc.transports' value changed : tcp 1048576 =>
udp 32768
Jul/26/2016 10:21:06 : Parameter 'sunrpc.transports' value changed : tcp 1048576 =>

```

```

tcp-bc 1048576
-bash-4.1# tfactl analyze
INFO: analyzing all (Alert and Unix System Logs) logs for the last 60 minutes...
Please wait...
INFO: analyzing host: myserver69

Report title: Analysis of Alert, System Logs
Report date range: last ~1 hour(s)
Report (default) time zone: UTC - Coordinated Universal Time
Analysis started at: 26-Jul-2016 10:36:03 AM UTC
Elapsed analysis time: 1 second(s).
Configuration file: /scratch/app/11.2.0.4/grid/tfa/myserver69/
tfa_home/ext/tnt/conf/tnt.prop
Configuration group: all
Total message count:          15,261, from 20-Nov-2015 02:06:21 AM
UTC to 26-Jul-2016 10:10:58 AM UTC
Messages matching last ~1 hour(s):          1, from 26-Jul-2016 10:10:58 AM
UTC to 26-Jul-2016 10:10:58 AM UTC
last ~1 hour(s) error count:                0
last ~1 hour(s) ignored error count:        0
last ~1 hour(s) unique error count:         0

Message types for last ~1 hour(s)
Occurrences percent  server name          type
-----
1 100.0% myserver69          generic
-----
1 100.0%

Unique error messages for last ~1 hour(s)
Occurrences percent  server name          error
-----
0 100.0%

```

tfactl events

Use the `tfactl events` command to view the events detected by Oracle Trace File Analyzer.

Syntax

```
tfactl events
```

Example

```

$ tfactl events
Output from host : myserver69
-----
Jul/25/2016 06:25:33 :
      [crs.myserver69] : [cssd(7513)]CRS-1603:CSSD on node myserver69 shutdown
by user.
Jul/25/2016 06:32:41 :
      [crs.myserver69] : [cssd(5794)]CRS-1601:CSSD Reconfiguration complete.
Active nodes are myserver69 myserver70 myserver71 .
Jul/25/2016 06:47:37 :
      [crs.myserver69] : [/scratch/app/11.2.0.4/grid/bin/scriptagent.bin(16233)]
CRS-5818:Aborted command 'start' for resource 'ora.oc4j'. Details at (:CRSAGF00113:)
{1:32892:193} in /scratch/app/11.2.0.4/grid/log/myserver69/agent/crsd/
scriptagent_oragrid/scriptagent_oragrid.log.

```


message and only shows those matching the last x number of minutes or any interval of time.

- Analyze the Oracle OSWatcher log statistics:** Oracle Trace File Analyzer reads the various statistics available in the OSWatcher log files and provides detailed analysis showing first, highest, lowest, average, and the last three readings of each statistic. Choose any interval down to a specific minute or second. Oracle Trace File Analyzer optionally provides the original data from the OSWatcher logs for each value reported on (data point).

Syntax

```
tfactl analyze [-search "pattern"] [-comp db | asm | crs | acfs | os | osw |
oswslabinfo | all]
[-type error | warning | generic] [-last nh[d]
[-from "MMM/DD/YYYY HH24:MI:SS"] [-to "MMM/DD/YYYY HH24:MI:SS"] [-for "MMM/DD/YYYY
HH24:MI:SS"]
[-node all | local | n1,n2,...] [-verbose] [-o file]
```

Parameters

Table 24 tfactl analyze Command Parameters

Parameter	Description
-search "pattern"	<p>Searches for a pattern enclosed in double quotation marks ("") in system and alert logs within a specified time range. This parameter supports both case-sensitive and case-insensitive search in alert and system message files across the cluster within the specified filters. Default is case insensitive.</p> <p>If you do not specify the -search parameter, then Oracle Trace File Analyzer provides a summary of messages within specified filters from alert and system log messages across the cluster.</p> <p>Oracle Trace File Analyzer displays message counts grouped by type (error, warning, and generic) and shows unique messages in a table organized by message type selected for analysis. The generic message type is assigned to all messages which are not either an error or warning message type.</p>
-comp db asm crs acfs os osw oswslabinfo all	<p>Select which components you want Oracle Trace File Analyzer to analyze. Default is all.</p> <ul style="list-style-type: none"> db: Database alert logs asm: Oracle ASM alert logs crs: Oracle Grid Infrastructure alert logs acfs: Oracle ACFS alert logs os: System message files osw: OSW Top output oswslabinfo: OSW Slabinfo output <p>When OSWatcher data is available, OSW and OSWLABINFO components provide summary views of OSWatcher data.</p>
-type error warning generic	Select what type of messages Oracle Trace File Analyzer analyzes. Default is error.
-last n[h d]	Specify an amount of time, in hours or days, before current time that you want Oracle Trace File Analyzer to analyze.

Table 24 (Cont.) tfactl analyze Command Parameters

Parameter	Description
-from -to -for "MMM/DD/YYYY HH24:MI:SS"	Specify a time interval, using the -from and -to parameters together, or a specific time using the -for parameter, that you want Oracle Trace File Analyzer to analyze.
-node all local n1,n2,...	Specify a comma-separated list of host names. Use -local to analyze files on the local node. Default is all.
-verbose	Displays verbose output.
-o file	Specify a file where Oracle Trace File Analyzer writes the output instead of displaying on the screen.

-type Parameter Arguments

The `tfactl analyze` command classifies all the messages into different categories when you specify the `-type` parameter. The analysis component provides count of messages by the message type you configure and lists all unique messages grouped by count within specified filters. The message type patterns for each argument are listed in the following table.

Table 25 tfactl analyze -type Parameter Arguments

Argument	Description
error	<p>Error message patterns for database and Oracle ASM alert logs:</p> <pre>.*ORA-00600:.* .*ORA-07445:.* .*IPC Send timeout detected. Sender: ospid.* .*Direct NFS: channel id .* path .* to filer .* PING timeout.* .*Direct NFS: channel id .* path .* to filer .* is DOWN.* .*ospid: .* has not called a wait for .* secs.* .*IPC Send timeout to .* inc .* for msg type .* from opid.* .*IPC Send timeout: Terminating pid.* .*Receiver: inst .* binc .* ospid.* .* terminating instance due to error.* .*: terminating the instance due to error.* .*Global Enqueue Services Deadlock detected</pre> <p>Error message patterns for Oracle Grid Infrastructure alert logs:</p> <pre>.*CRS-8011:.*,.*CRS-8013:.*,.*CRS-1607:.*,.*CRS-1615:.*, .*CRS-1714:.*,.*CRS-1656:.*,.*PRVF-5305:.*,.*CRS-1601:.*, .*CRS-1610:.*,.*PANIC. CRSD exiting:.*,.*Fatal Error from AGFW Proxy:.*</pre>
warning	<p>Warning message patterns for database and Oracle ASM alert logs:</p> <pre>NOTE: process .* initiating offline of disk .* .*WARNING: cache read a corrupted block group.* .*NOTE: a corrupted block from group FRA was dumped to</pre>
generic	Any messages that do not match any of the preceding patterns.

Examples

The following command examples demonstrate how to use Oracle Trace File Analyzer to search collected data:

- `$ tfactl analyze -search "error" -last 2d`
Oracle Trace File Analyzer searches alert and system log files from the past two days for messages that contain the case-insensitive string "error".
- `$ tfactl analyze -comp os -for "Jul/01/2016 11" -search "."`
Oracle Trace File Analyzer displays all system log messages for July 1, 2016 at 11 am.
- `$ tfactl analyze -search "/ORA-/c" -comp db -last 2d`
Oracle Trace File Analyzer searches database alert logs for the case-sensitive string "ORA-" from the past two days.

The following command examples demonstrate how to use Oracle Trace File Analyzer to analyze collected data:

- `$ tfactl analyze -last 5h`
Oracle Trace File Analyzer displays a summary of events collected from all alert logs and system messages from the past five hours.
- `$ tfactl analyze -comp os -last 1d`
Oracle Trace File Analyzer displays a summary of events from system messages from the past day.
- `$ tfactl analyze -last 1h -type generic`
Oracle Trace File Analyzer analyzes all generic messages from the last hour.

The following command examples demonstrate how to use Oracle Trace File Analyzer to analyze OSWatcher Top and Slabinfo:

- `$ tfactl analyze -comp osw -last 6h`
Oracle Trace File Analyzer displays OSWatcher Top summary for the past six hours.
- `$ tfactl analyze -comp oswslabinfo -from "2016-07-01" -to "2016-07-03"`
Oracle Trace File Analyzer displays OSWatcher Slabinfo summary for specified time period.

tfactl run

Use the `tfactl run` command to run the requested action (can be inventory or scan or any support tool).

Syntax

```
tfactl run [inventory | scan | tool]
```

Parameters

Table 26 `tfactl run` Command Parameters

Parameter	Description
<code>inventory</code>	Inventory of all trace file directories.
<code>scan</code>	Runs a one off scan.
<code>tool</code>	Runs the desired analysis tool.

Analysis Tools

Table 27 `tfactl run` Analysis Tools Parameters

Parameter	Description
<code>changes</code>	Prints system changes.
<code>events</code>	Lists all important events in system.
<code>exachk</code>	Runs Oracle EXAchk.
<code>grep</code>	<code>grep</code> for input string in logs.
<code>history</code>	Lists commands run in current Oracle Trace File Analyzer shell session.
<code>ls</code>	Searches files in Oracle Trace File Analyzer.
<code>orachk</code>	Runs Oracle ORAchk.
<code>oratop</code>	Runs <code>oratop</code> .
<code>oswbb</code>	Runs OSWatcher Analyzer.
<code>param</code>	Prints parameter value.
<code>ps</code>	Finds a process.
<code>pstack</code>	Runs <code>pstack</code> on a process.
<code>prw</code>	Runs Procwatcher.
<code>sqlt</code>	Runs SQLT.
<code>summary</code>	Prints system summary.
<code>tail</code>	Tails log files.
<code>vi</code>	Searches and opens files in the <code>vi</code> editor.

Profiling Tools

Table 28 `tfactl run` Profiling Tools Parameters

Parameter	Description
<code>dbglevel</code>	Sets CRS log and trace levels using profiles.

`tfactl toolstatus`

Use the `tfactl toolstatus` command to view the status of Oracle Trace File Analyzer Support Tools across all nodes.

Syntax

```
$ tfactl toolstatus
```

Example

The `tfactl toolstatus` command returns output similar to the following, showing which tool is deployed and where the tool is deployed.

Table 29 tfactl toolstatus Output

Host	Tool	Status
hostname	alertsummary	DEPLOYED
hostname	exachk	DEPLOYED
hostname	ls	DEPLOYED
hostname	trriage	DEPLOYED
hostname	pstack	DEPLOYED
hostname	orachk	DEPLOYED
hostname	sqlt	DEPLOYED
hostname	grep	DEPLOYED
hostname	summary	DEPLOYED
hostname	vi	DEPLOYED
hostname	prw	NOT RUNNING
hostname	tail	DEPLOYED
hostname	param	DEPLOYED
hostname	dbglevel	DEPLOYED
hostname	managelogs	DEPLOYED
hostname	history	DEPLOYED
hostname	oratop	DEPLOYED
hostname	calog	DEPLOYED
hostname	menu	DEPLOYED
hostname	oswbb	RUNNING
hostname	changes	DEPLOYED
hostname	events	DEPLOYED
hostname	ps	DEPLOYED
hostname	srdc	DEPLOYED

Running Diagnostic Collection Commands

Run the diagnostic collection commands to collect diagnostic data.

- [tfactl diagcollect](#)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

- **tfactl directory**
Use the `tfactl directory` command to add a directory to, or remove a directory from the list of directories to analyze their trace or log files.
- **tfactl ips**
Use the `tfactl ips` command to collect Automatic Diagnostic Repository diagnostic data.
- **tfactl collection**
Use the `tfactl collection` command to stop a running Oracle Trace File Analyzer collection.
- **tfactl print**
Use the `tfactl print` command to print information from the Berkeley database.
- **tfactl purge**
Use the `tfactl purge` command to delete diagnostic collections from the Oracle Trace File Analyzer repository that are older than a specific time.
- **tfactl managelogs**
Use the `tfactl managelogs` command to manage Automatic Diagnostic Repository log and trace files.

tfactl diagcollect

Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

Oracle Trace File Analyzer Collector can perform three types of on-demand collections:

- Default collections
- Event-driven Support Service Request Data Collection (SRDC) collections
- Custom collections

Prerequisites

Event-driven Support Service Request Data Collection (SRDC) collections require components from the Oracle Trace File Analyzer Database Support Tools Bundle, which is available from My Oracle Support Note 1513912.2.

Syntax

```
tfactl diagcollect [-all | [component_name1] [component_name2] ...
[component_nameN]]
[-node all|local|n1,n2,..] [-tag description]
[-z filename]
[-last nh|d] [-from time -to time | -for time]
[-nocopy] [-notrim] [-silent] [-nocores] [-collectalldirs]
[-collectdir dir1,dir2..] [-examples]
[-node [node1,node2,nodeN]]
components:-ips|-database|-asm|-crsclient|-dbclient|-dbwlm|-tns|-rhp|-procinfo|-afd
|-crs|-wls|-emagent|-oms|-ocm|-emplugins|-em|-acfs|-install|-cfgtools|-os|-ips
|-ashhtml|-ashtext|-awrhtml|-awrtext
```

Parameters

Each option must be prefixed with a minus sign (-).

Option	Description
-all [component_name1] [component_name2] ... [component_nameN]	Specify that you want to collect data on all components, or specify specific components for which you want to obtain collections.
-node all local n1,n2,...	Specify a comma-delimited list of nodes from which to collect diagnostic information. Default is all.
-tag <i>description</i>	Use this parameter to create a subdirectory for the resulting collection in the Oracle Trace File Analyzer repository.
-z <i>file_name</i>	Use this parameter to specify an output file name.
-last <i>number</i> h d - from " <i>mmm/dd/yyyy</i> <i>hh:mm:ss</i> " -to " <i>mmm/dd/</i> <i>yyyy hh:mm:ss</i> " -for " <i>mmm/dd/yyyy hh:mm:ss</i> "	<ul style="list-style-type: none"> Specify the <code>-last</code> parameter to collect files that have relevant data for the past specific number of hours (<i>h</i>) or days (<i>d</i>). By default, using the command with this parameter also trims files that are large and shows files only from the specified interval. Specify the <code>-from</code> and <code>-to</code> parameters (you must use these two parameters together) to collect files that have relevant data during a specific time interval, and trim data before this time where files are large. Specify the <code>-for</code> parameter to collect files that have relevant data for the time given. The files TFACTL collects will have timestamps in between which the time you specify after <code>-for</code> is included. No data trimming is done for this option.
-nocopy	Specify this parameter to stop the resultant trace file collection from being copied back to the initiating node. The file remains in the Oracle Trace File Analyzer repository on the executing node.
-notrim	Specify this parameter to stop trimming the files collected.
-silent	Specify this parameter to run diagnostic collection as a background process
-nocores	Specify this parameter to stop collecting core files when it would normally have been collected.
-collectalldirs	Specify this parameter to collect all files from a directory that has Collect All flag marked true.
-collectdir <i>dir1,dir2,...dirn</i>	Specify a comma-delimited list of directories and collection includes all files from these directories irrespective of type and time constraints in addition to the components specified.
-examples	Specify this parameter to view <code>diagcollect</code> usage examples.

 **Note:**

If you specify both date and time, then you must enclose both the values in double quotation marks (""). If you specify only the date or the time, then you do not have to enclose the single value in quotation marks.

Examples

- The following command trims and zips all files updated in the last four hours, including `chmos` and `osw` data, from across the cluster and collects it on the initiating node:

```
$ tfactl diagcollect -all

Collecting data for the last 12 hours for this component ...
Collecting data for all nodes
Creating ips package in master node ...
Trying ADR basepath /scratch/app/orabase
Trying to use ADR homepath diag/crs/node1/crs ...
Submitting request to generate package for ADR homepath /scratch/app/orabase/
diag/crs/node1/crs
Trying ADR basepath /scratch/app/oracle
Trying to use ADR homepath diag/rdbms/prod/prod_1 ...
Submitting request to generate package for ADR homepath /scratch/app/oracle/diag/
rdbms/prod/prod_1
Trying to use ADR homepath diag/rdbms/prod/prod_2 ...
Submitting request to generate package for ADR homepath /scratch/app/oracle/diag/
rdbms/prod/prod_2
Trying to use ADR homepath diag/rdbms/webdb/webdb_2 ...
Submitting request to generate package for ADR homepath /scratch/app/oracle/diag/
rdbms/webdb/webdb_2
Trying to use ADR homepath diag/rdbms/webdb/webdb_1 ...
Submitting request to generate package for ADR homepath /scratch/app/oracle/diag/
rdbms/webdb/webdb_1
Master package completed for ADR homepath /scratch/app/oracle/diag/rdbms/prod/
prod_1
Master package completed for ADR homepath /scratch/app/oracle/diag/rdbms/prod/
prod_2
Master package completed for ADR homepath /scratch/app/oracle/diag/rdbms/webdb/
webdb_1
Master package completed for ADR homepath /scratch/app/oracle/diag/rdbms/webdb/
webdb_2
Master package completed for ADR homepath /scratch/app/orabase/diag/crs/node1/crs
Created package 2 based on time range 2016-09-29 12:11:00.000000 -07:00 to
2016-09-30 00:11:00.000000 -07:00,
correlation level basic
Remote package completed for ADR homepath(s) /diag/crs/node2/crs,/diag/crs/node3/
crs

Collection Id : 20160930001113node1

Detailed Logging at : /scratch/app/orabase/tfa/repository/
collection_Fri_Sep_30_00_11_13_PDT_2016_node_all/
diagcollect_20160930001113_node1.log
2016/09/30 00:12:21 PDT : Collection Name : tfa_Fri_Sep_30_00_11_13_PDT_2016.zip
2016/09/30 00:12:21 PDT : Collecting diagnostics from hosts : [node1, node3,
node2]
2016/09/30 00:12:21 PDT : Scanning of files for Collection in progress...
2016/09/30 00:12:21 PDT : Collecting additional diagnostic information...
2016/09/30 00:12:26 PDT : Getting list of files satisfying time range
[09/29/2016 12:12:21 PDT, 09/30/2016 00:12:26 PDT]
2016/09/30 00:13:05 PDT : Collecting ADR incident files...
2016/09/30 00:15:02 PDT : Completed collection of additional diagnostic
information...
2016/09/30 00:15:24 PDT : Completed Local Collection
2016/09/30 00:15:26 PDT : Remote Collection in Progress...
```

Collection Summary			
Host	Status	Size	Time
node3	Completed	82MB	172s
node2	Completed	95MB	183s
node1	Completed	157MB	183s

```

Logs are being collected to: /scratch/app/orabase/tfa/repository/
collection_Fri_Sep_30_00_11_13_PDT_2016_node_all
/scratch/app/orabase/tfa/repository/
collection_Fri_Sep_30_00_11_13_PDT_2016_node_all/
node3.tfa_Fri_Sep_30_00_11_13_PDT_2016.zip
/scratch/app/orabase/tfa/repository/
collection_Fri_Sep_30_00_11_13_PDT_2016_node_all/
node2.tfa_Fri_Sep_30_00_11_13_PDT_2016.zip
/scratch/app/orabase/tfa/repository/
collection_Fri_Sep_30_00_11_13_PDT_2016_node_all/
node1.tfa_Fri_Sep_30_00_11_13_PDT_2016.zip

```

- The following command trims and zips all files updated in the last eight hours, including `chmos` and `osw` data, from across the cluster and collects it on the initiating node:

```
$ tfactl diagcollect -all -last 8h
```

- The following command trims and zips all files from databases `hrdb` and `fdb` updated in the last one day and collects it on the initiating node:

```
$ tfactl diagcollect -database hrdb,fdb -last 1d -z foo
```

- The following command trims and zips all Oracle Clusterware files, operating system logs, and `chmos` and `osw` data from `node1` and `node2` updated in the last six hours, and collects it on the initiating node:

```
$ tfactl diagcollect -crs -os -node node1,node2 -last 6h
```

- The following command trims and zips all Oracle ASM logs from `node1` updated between September 22, 2016 and September 23, 2016 at 21:00, and collects it on the initiating node:

```
$ tfactl diagcollect -asm -node node1 -from Sep/22/2016 -to "Sep/23/2016
21:00:00"
```

- The following command trims and zips all log files updated on September 23, 2016 and collect at the initiating node:

```
$ tfactl diagcollect -for Sep/23/2016
```

- The following command trims and zips all log files updated from 09:00 on September 22, 2016, to 09:00 on September 23, 2016, which is 12 hours before and after the time specified in the command, and collects it on the initiating node:

```
$ tfactl diagcollect -for "September/22/2016 21:00:00"
```

Related Topics

- <https://support.oracle.com/rs?type=doc&id=1513912.2>

tfactl directory

Use the `tfactl directory` command to add a directory to, or remove a directory from the list of directories to analyze their trace or log files.

Also, use the `tfactl directory` command to change the directory permissions. When automatic discovery adds a directory, the directory is added as public. Any user who has sufficient permissions to run the `tfactl diagcollect` command collects any file in that directory. This is only important when non-root or `sudo` users run TFACTL commands.

If a directory is marked as private, then Oracle Trace File Analyzer, before allowing any files to be collected:

- Determines which user is running TFACTL commands
- Verifies if the user has permissions to see the files in the directory



Note:

A user can only add a directory to Oracle Trace File Analyzer to which they have read access. If you have automatic diagnostic collections configured, then Oracle Trace File Analyzer runs as `root`, and can collect all available files.

The `tfactl directory` command includes three verbs with which you can manage directories: `add`, `remove`, and `modify`.

Syntax

```
tfactl directory add directory [-public] [-exclusions | -noexclusions | -collectall] [-node all | n1,n2...]
```

```
tfactl directory remove directory [-node all | n1,n2...]
```

```
tfactl directory modify directory [-private | -public] [-exclusions | -noexclusions | -collectall]
```

For each of the three syntax models, you must specify a directory path where Oracle Trace File Analyzer stores collections.

Parameters

Table 30 tfactl directory Command Parameters

Parameter	Description
<code>-public</code>	Use the <code>-public</code> parameter to make the files contained in the directory available for collection by any Oracle Trace File Analyzer user.

Table 30 (Cont.) tfactl directory Command Parameters

Parameter	Description
-private	Use the <code>-private</code> parameter to prevent an Oracle Trace File Analyzer user who does not have permission to see the files in a directory (and any subdirectories) you are adding or modifying, from running a command to collect files from the specified directory.
-exclusions	Use the <code>-exclusions</code> parameter to specify that files in this directory are eligible for collection if the files satisfy type, name, and time range restrictions.
-noexclusions	Use the <code>-noexclusions</code> parameter to specify that files in this directory are eligible for collection if the files satisfy time range restrictions.
-collectall	Use the <code>-collectall</code> parameter to specify that files in this directory are eligible for collection irrespective of type and time range when the user specifies the <code>-collectalldirs</code> parameter with the <code>tfactl diagcollect</code> command.
-node all n1,n2...	Add or remove directories from every node in the cluster or use a comma-delimited list to add or remove directories from specific nodes.

Usage Notes

You must add all trace directory names to the Berkeley database so that Oracle Trace File Analyzer can collect file metadata in that directory. The discovery process finds most directories, but if new or undiscovered directories are required, then you can add these manually using the `tfactl directory` command.

When you add a directory using TFACTL, then Oracle Trace File Analyzer attempts to determine whether the directory is for

- Oracle database
- Oracle Clusterware
- Operating system logs
- Some other component
- Which database or instance

If Oracle Trace File Analyzer cannot determine this information, then Oracle Trace File Analyzer returns an error and requests that you enter the information, similar to the following:

```
# tfactl directory add /tmp
```

```
Failed to add directory to TFA. Unable to determine parameters for directory: /tmp
Please enter component for this Directory [RDBMS|CRS|ASM|INSTALL|OS|CFGTOOLS|TNS|
DBWLM|ACFS|ALL] : RDBMS
Please enter database name for this Directory :MYDB
Please enter instance name for this Directory :MYDB1
```

**Note:**

For OS, CRS, CFGTOOLS, ACFS, ALL, or INSTALL files, only the component is requested and for Oracle ASM only the instance is created. No verification is done for these entries so use caution when entering this data.

Examples

The following command adds a directory:

```
# tfactl directory add /u01/app/grid/diag/asm/+ASML/trace
```

The following command modifies a directory and makes the contents available for collection only to Oracle Trace File Analyzer users with sufficient permissions:

```
# tfactl directory modify /u01/app/grid/diag/asm/+ASML/trace -private
```

The following command removes a directory from all nodes in the cluster:

```
# tfactl directory remove /u01/app/grid/diag/asm/+ASML/trace -node all
```

tfactl ips

Use the `tfactl ips` command to collect Automatic Diagnostic Repository diagnostic data.

Syntax

```
tfactl ips [ADD] [ADD FILE] [ADD NEW INCIDENTS] [CHECK REMOTE KEYS] [COPY IN FILE]
[ COPY OUT FILE] [CREATE PACKAGE] [DELETE PACKAGE] [FINALIZE PACKAGE] [GENERATE
PACKAGE]
[GET MANIFEST] [GET METADATA] [GET REMOTE KEYS] [PACK] [REMOVE] [REMOVE FILE]
[SET CONFIGURATION] [SHOW CONFIGURATION] [SHOW FILES] [SHOW INCIDENTS] [SHOW
PROBLEMS]
[SHOW PACKAGE] [UNPACK FILE] [UNPACK PACKAGE] [USE REMOTE KEYS] [options]
```

Parameters**Table 31 tfactl ips Command Parameters**

Parameter	Description
ADD	Adds incidents to an existing package.
ADD FILE	Adds a file to an existing package.
ADD NEW INCIDENTS	Finds new incidents for the problems and add the latest ones to the package.
CHECK REMOTE KEYS	Creates a file with keys matching incidents in specified package.
COPY IN FILE	Copies an external file into Automatic Diagnostic Repository, and associates it with a package and (optionally) an incident.
COPY OUT FILE	Copies an Automatic Diagnostic Repository file to a location outside Automatic Diagnostic Repository.
CREATE PACKAGE	Creates a package, and optionally select contents for the package.

Table 31 (Cont.) tfactl ips Command Parameters

Parameter	Description
DELETE PACKAGE	Drops a package and its contents from Automatic Diagnostic Repository.
FINALIZE PACKAGE	Gets a package ready for shipping by automatically including correlated contents.
GENERATE PACKAGE	Creates a physical package (<code>zip</code> file) in target directory.
GET MANIFEST	Extracts the manifest from a package file and displays it.
GET METADATA	Extracts the metadata XML document from a package file and displays it.
GET REMOTE KEYS	Creates a file with keys matching incidents in specified package.
PACK	Creates a package, and immediately generates the physical package.
REMOVE	Removes incidents from an existing package.
REMOVE FILE	Removes a file from an existing package.
SET CONFIGURATION	Changes the value of an Incident Packaging Service configuration parameter.
SHOW CONFIGURATION	Shows the current Incident Packaging Service settings.
SHOW FILES	Shows the files included in the specified package.
SHOW INCIDENTS	Shows incidents included in the specified package.
SHOW PROBLEMS	Shows problems for the current Automatic Diagnostic Repository home.
SHOW PACKAGE	Shows details for the specified package.
UNPACK FILE	Unpackages a physical file into the specified path.
UNPACK PACKAGE	Unpackages physical files in the current directory into the specified path, if they match the package name.
USE REMOTE KEYS	Adds incidents matching the keys in the specified file to the specified package.

- [tfactl ips ADD](#)
Use the `tfactl ips ADD` command to add incidents to an existing package.
- [tfactl ips ADD FILE](#)
Use the `tfactl ADD FILE` command to add a file to an existing package.
- [tfactl ips COPY IN FILE](#)
Use the `tfactl ips COPY IN FILE` command to copy an external file into Automatic Diagnostic Repository, and associate the file with a package and (optionally) an incident.
- [tfactl ips REMOVE](#)
Use the `tfactl ips REMOVE` command to remove incidents from an existing package.
- [tfactl ips REMOVE FILE](#)
Use the `tfactl ips REMOVE FILE` command to remove a file from an existing package.

- **tfactl ips ADD NEW INCIDENTS PACKAGE**
Use the `tfactl ips ADD NEW INCIDENTS PACKAGE` command to find new incidents for the problems in a specific package, and add the latest ones to the package.
- **tfactl ips GET REMOTE KEYS FILE**
Use the `tfactl ips GET REMOTE KEYS FILE` command to create a file with keys matching incidents in a specific package.
- **tfactl ips USE REMOTE KEYS FILE**
Use the `tfactl ips USE REMOTE KEYS FILE` command to add incidents matching the keys in a specific file to a specific package.
- **tfactl ips CREATE PACKAGE**
Use the `tfactl ips CREATE PACKAGE` command to create a package, and optionally select the contents for the package.
- **tfactl ips FINALIZE PACKAGE**
Use the `tfactl ips FINALIZE PACKAGE` command to get a package ready for shipping by automatically including correlated contents.
- **tfactl ips GENERATE PACKAGE**
Use the `tfactl ips GENERATE PACKAGE` command to create a physical package (zip file) in the target directory.
- **tfactl ips DELETE PACKAGE**
Use the `tfactl ips DELETE PACKAGE` command to drop a package and its contents from the Automatic Diagnostic Repository.
- **tfactl ips GET MANIFEST FROM FILE**
Use the `tfactl ips GET MANIFEST FROM FILE` command to extract the manifest from a package file and view it.
- **tfactl ips GET METADATA**
Use the `tfactl ips GET METADATA` command to extract the metadata XML document from a package file and view it.
- **tfactl ips PACK**
Use the `tfactl ips PACK` command to create a package and immediately generate the physical package.
- **tfactl ips SET CONFIGURATION**
Use the `tfactl ips SET CONFIGURATION` command to change the value of an Incident Packaging Service configuration parameter.
- **tfactl ips SHOW CONFIGURATION**
Use the `tfactl ips SHOW CONFIGURATION` command to view the current Incident Packaging Service settings.
- **tfactl ips SHOW PACKAGE**
Use the `tfactl ips SHOW PACKAGE` command to view the details of a specific package.
- **tfactl ips SHOW FILES PACKAGE**
Use the `tfactl ips SHOW FILES PACKAGE` command to view the files included in a specific package.
- **tfactl ips SHOW INCIDENTS PACKAGE**
Use the `tfactl ips SHOW INCIDENTS PACKAGE` command to view the incidents included in a specific package.

- [tfactl ips SHOW PROBLEMS](#)
Use the `tfactl ips SHOW PROBLEMS` command to view the problems for the current Automatic Diagnostic Repository home.
- [tfactl ips UNPACK FILE](#)
Use the `tfactl ips UNPACK FILE` command to unpack a physical file into a specific path.
- [tfactl ips UNPACK PACKAGE](#)
Use the `tfactl ips UNPACK PACKAGE` command to unpack physical files in the current directory into a specific path, if they match the package name.

tfactl ips ADD

Use the `tfactl ips ADD` command to add incidents to an existing package.

Syntax

```
tfactl ips ADD [INCIDENT incid | PROBLEM prob_id | PROBLEMKEY prob_key | SECONDS
seconds | TIME start_time TO end_time] PACKAGE package_id
```

Parameters

Table 32 tfactl ips ADD Command Parameters

Parameter	Description
<i>incid</i>	Specify the ID of the incident to add to the package contents.
<i>prob_id</i>	Specify the ID of the problem to add to the package contents.
<i>prob_key</i>	Specify the problem key to add to the package contents.
<i>seconds</i>	Specify the number of seconds before now for adding package contents.
<i>start_time</i>	Specify the start of time range to look for incidents in.
<i>end_time</i>	Specify the end of time range to look for incidents in.

tfactl ips ADD FILE

Use the `tfactl ADD FILE` command to add a file to an existing package.

Syntax

The file must be in the same `ADR_BASE` as the package.

```
tfactl ips ADD FILE file_spec PACKAGE pkgid
```

Parameters

Table 33 tfactl ips ADD FILE Command Parameters

Parameter	Description
<i>file_spec</i>	Specify the file with file and path (full or relative).
<i>package_id</i>	Specify the ID of the package to add the file to.

tfactl ips COPY IN FILE

Use the `tfactl ips COPY IN FILE` command to copy an external file into Automatic Diagnostic Repository, and associate the file with a package and (optionally) an incident.

Syntax

```
tfactl ips COPY IN FILE file [TO new_name] [OVERWRITE] PACKAGE pkgid [INCIDENT incid]
```

Parameters

Table 34 tfactl ips COPY IN FILE Command Parameters

Parameter	Description
<i>file</i>	Specify the file with file name and full path (full or relative).
<i>new_name</i>	Specify a name for the copy of the file.
<i>pkgid</i>	Specify the ID of the package to associate the file with.
<i>incid</i>	Specify the ID of the incident to associate the file with.

Options

OVERWRITE: If the file exists, then use the OVERWRITE option to overwrite the file.

tfactl ips REMOVE

Use the `tfactl ips REMOVE` command to remove incidents from an existing package.

Syntax

The incidents remain associated with the package, but not included in the physical package file.

```
tfactl ips REMOVE [INCIDENT incid | PROBLEM prob_id | PROBLEMKEY prob_key] PACKAGE package_id
```

Parameters

Table 35 tfactl ips REMOVE Command Parameters

Parameter	Description
<i>incid</i>	Specify the ID of the incident to add to the package contents.
<i>prob_id</i>	Specify the ID of the problem to add to the package contents.
<i>prob_key</i>	Specify the problem key to add to the package contents.

Example

```
$ tfactl ips remove incident 22 package 12
```

tfactl ips REMOVE FILE

Use the `tfactl ips REMOVE FILE` command to remove a file from an existing package.

Syntax

The file must be in the same `ADR_BASE` as the package. The file remains associated with the package, but not included in the physical package file.

```
tfactl ips REMOVE FILE file_spec PACKAGE pkgid
```

Example

```
$ tfactl ips remove file ADR_HOME/trace/mydb1_ora_13579.trc package 12
```

tfactl ips ADD NEW INCIDENTS PACKAGE

Use the `tfactl ips ADD NEW INCIDENTS PACKAGE` command to find new incidents for the problems in a specific package, and add the latest ones to the package.

Syntax

```
tfactl ips ADD NEW INCIDENTS PACKAGE package_id
```

Parameters

Table 36 tfactl ips ADD NEW INCIDENTS PACKAGE Command Parameters

Parameter	Description
<i>package_id</i>	Specify the ID of the package to add the incidents to.

tfactl ips GET REMOTE KEYS FILE

Use the `tfactl ips GET REMOTE KEYS FILE` command to create a file with keys matching incidents in a specific package.

Syntax

```
tfactl ips GET REMOTE KEYS FILE file_spec PACKAGE package_id
```

Parameters

Table 37 tfactl ips GET REMOTE KEYS FILE Command Parameters

Parameter	Description
<i>file_spec</i>	Specify the file with file name and full path (full or relative).
<i>package_id</i>	Specify the ID of the package to get keys for.

Example

```
$ tfactl ips get remote keys file /tmp/key_file.txt package 12
```

tfactl ips USE REMOTE KEYS FILE

Use the `tfactl ips USE REMOTE KEYS FILE` command to add incidents matching the keys in a specific file to a specific package.

Syntax

```
tfactl ips USE REMOTE KEYS FILE file_spec PACKAGE package_id
```

Example

```
$ tfactl ips use remote keys file /tmp/key_file.txt package 12
```

tfactl ips CREATE PACKAGE

Use the `tfactl ips CREATE PACKAGE` command to create a package, and optionally select the contents for the package.

Syntax

```
tfactl ips CREATE PACKAGE [INCIDENT inc_id | PROBLEM prob_id  
| PROBLEMKEY prob_key | SECONDS seconds | TIME start_time TO end_time] [CORRELATE  
BASIC | TYPICAL | ALL] [MANIFEST file_spec]  
[KEYFILE file_spec]
```

Parameters

Table 38 tfactl ips CREATE PACKAGE Command Parameters

Parameter	Description
<i>inc_id</i>	Specify the ID of the incident to use for selecting the package contents.
<i>prob_id</i>	Specify the ID of the problem to use for selecting the package contents.
<i>prob_key</i>	Specify the problem key to use for selecting the package contents.
<i>seconds</i>	Specify the number of seconds before now for selecting the package contents.
<i>start_time</i>	Specify the start of time range to look for the incidents in.
<i>end_time</i>	Specify the end of time range to look for the incidents in.

Options

- `CORRELATE BASIC`: The package includes the incident dumps and the incident process trace files. If the incidents share relevant correlation keys, then more incidents are included automatically.
- `CORRELATE TYPICAL`: The package includes the incident dumps and all trace files that were modified in a time window around each incident. If the incidents share relevant correlation keys, or occurred in a time window around the main incidents, then more incidents are included automatically.
- `CORRELATE ALL`: The package includes the incident dumps and all trace files that were modified between the first selected incident and the last selected incident. If

the incidents occurred in the same time range, then more incidents are included automatically.

- `MANIFEST file_spec`: Generates the XML format package manifest file.
- `KEYFILE file_spec`: Generates the remote key file.

 **Note:**

- If you do not specify package contents, such as incident, problem, and so on, then Oracle Trace File Analyzer creates an empty package. You can add files and incidents later.
- If you do not specify the correlation level, then Oracle Trace File Analyzer uses the default level.
- The default is normally **TYPICAL**, but you can change using the `IPS SET CONFIGURATION` command.

Example

```
$tfactl ips create package incident 861
```

```
$ tfactl ips create package time '2006-12-31 23:59:59.00 -07:00' to '2007-01-01 01:01:01.00 -07:00'
```

tfactl ips FINALIZE PACKAGE

Use the `tfactl ips FINALIZE PACKAGE` command to get a package ready for shipping by automatically including correlated contents.

Syntax

```
tfactl ips FINALIZE PACKAGE package_id
```

tfactl ips GENERATE PACKAGE

Use the `tfactl ips GENERATE PACKAGE` command to create a physical package (zip file) in the target directory.

Syntax

```
tfactl ips GENERATE PACKAGE package_id [IN path][COMPLETE | INCREMENTAL]
```

Parameters

Table 39 tfactl ips GENERATE PACKAGE Command Parameters

Parameter	Description
<i>package_id</i>	Specify the ID of the package to create physical package file for.
<i>path</i>	Specify the path where the physical package file must be generated.

Options

- **COMPLETE:** (Default) The package includes all package files even if a previous package sequence was generated.
- **INCREMENTAL:** The package includes only the files that have been added or changed since the last package was generated.



Note:

If no target path is specified, then Oracle Trace File Analyzer generates the physical package file in the current working directory.

Example

```
$ tfactl ips generate package 12 in /tmp
```

tfactl ips DELETE PACKAGE

Use the `tfactl ips DELETE PACKAGE` command to drop a package and its contents from the Automatic Diagnostic Repository.

Syntax

```
tfactl ips DELETE PACKAGE package_id
```

Parameters

Table 40 tfactl ips DELETE PACKAGE Command Parameters

Parameter	Description
<i>package_id</i>	Specify the ID of the package to delete.

Example

```
$ tfactl ips delete package 12
```

tfactl ips GET MANIFEST FROM FILE

Use the `tfactl ips GET MANIFEST FROM FILE` command to extract the manifest from a package file and view it.

Syntax

```
tfactl ips GET MANIFEST FROM FILE file
```

Parameters

Table 41 tfactl ips GET MANIFEST FROM FILE Command Parameters

Parameter	Description
<i>file</i>	Specify the external file with file name and full path.

Example

```
$ tfactl ips GET MANIFEST FROM FILE /tmp/IPSPKG_200704130121_COM_1.zip
```

tfactl ips GET METADATA

Use the `tfactl ips GET METADATA` command to extract the metadata XML document from a package file and view it.

Syntax

```
tfactl ips GET METADATA [FROM FILE file | FROM ADR]
```

Example

```
$ tfactl ips get metadata from file /tmp/IPSPKG_200704130121_COM_1.zip
```

tfactl ips PACK

Use the `tfactl ips PACK` command to create a package and immediately generate the physical package.

Syntax

```
tfactl ips PACK [INCIDENT incid | PROBLEM prob_id | PROBLEMKEY prob_key | SECONDS  
seconds | TIME start_time TO end_time]  
[CORRELATE BASIC | TYPICAL | ALL] [MANIFEST file_spec] [KEYFILE file_spec]
```

Parameters

Table 42 tfactl ips PACK Command Parameters

Parameter	Description
<i>incid</i>	Specify the ID of the incident to use for selecting the package contents.
<i>prob_id</i>	Specify the ID of the problem to use for selecting the package contents.
<i>prob_key</i>	Specify the problem key to use for selecting the package contents.
<i>seconds</i>	Specify the number of seconds before the current time for selecting the package contents.
<i>start_time</i>	Specify the start of time range to look for the incidents in.
<i>end_time</i>	Specify the end of time range to look for the incidents in.
<i>path</i>	Specify the path where the physical package file must be generated.

Options

- **CORRELATE BASIC:** The package includes the incident dumps and the incident process trace files. If the incidents share relevant correlation keys, then more incidents are included automatically.
- **CORRELATE TYPICAL:** The package includes the incident dumps and all trace files that were modified in a time window around each incident. If the incidents share relevant correlation keys, or occurred in a time window around the main incidents, then more incidents are included automatically.
- **CORRELATE ALL:** The package includes the incident dumps and all trace files that were modified between the first selected incident and the last selected incident. If the incidents occurred in the same time range, then more incidents are included automatically.
- **MANIFEST file_spec:** Generate the XML format package manifest file.
- **KEYFILE file_spec:** Generate remote key file.

Note:

If you do not specify package contents, such as incident, problem, and so on, then Oracle Trace File Analyzer creates an empty package.

You can add files and incidents later.

If you do not specify the correlation level, then Oracle Trace File Analyzer uses the default level.

The default is normally **TYPICAL**, but you can change using the `IPS SET CONFIGURATION` command.

Example

```
$ tfactl ips pack incident 861
```

```
$ tfactl ips pack time '2006-12-31 23:59:59.00 -07:00' to '2007-01-01 01:01:01.00 -07:00'
```

tfactl ips SET CONFIGURATION

Use the `tfactl ips SET CONFIGURATION` command to change the value of an Incident Packaging Service configuration parameter.

Syntax

```
tfactl ips SET CONFIGURATION parameter_id value
```

Parameters

Table 43 tfactl ips SET CONFIGURATION Command Parameters

Parameter	Description
<i>parameter_id</i>	Specify the ID of the parameter to change.
<i>value</i>	Specify the new value for the parameter.

Example

```
$ tfactl ips set configuration 6 2
```

tfactl ips SHOW CONFIGURATION

Use the `tfactl ips SHOW CONFIGURATION` command to view the current Incident Packaging Service settings.

Syntax

```
tfactl ips SHOW CONFIGURATION parameter_id
```

tfactl ips SHOW PACKAGE

Use the `tfactl ips SHOW PACKAGE` command to view the details of a specific package.

Syntax

```
tfactl ips SHOW PACKAGE package_id [BASIC | BRIEF | DETAIL]
```

Note:

It is possible to specify the level of detail to use with this command.

BASIC : Shows a minimal amount of information. It is the default when no package ID is specified.

BRIEF : Shows a more extensive amount of information. It is the default when a package ID is specified.

DETAIL : Shows the same information as **BRIEF**, and also some package history and information on included incidents and files.

Example

```
$ tfactl ips show package  
$ tfactl ips show package 12 detail
```

tfactl ips SHOW FILES PACKAGE

Use the `tfactl ips SHOW FILES PACKAGE` command to view the files included in a specific package.

Syntax

```
tfactl ips SHOW FILES PACKAGE package_id
```

Example

```
$ tfactl ips show files package 12
```

tfactl ips SHOW INCIDENTS PACKAGE

Use the `tfactl ips SHOW INCIDENTS PACKAGE` command to view the incidents included in a specific package.

Syntax

```
tfactl ips SHOW INCIDENTS PACKAGE package_id
```

Example

```
$ tfactl ips show incidents package 12
```

tfactl ips SHOW PROBLEMS

Use the `tfactl ips SHOW PROBLEMS` command to view the problems for the current Automatic Diagnostic Repository home.

Syntax

```
tfactl ips SHOW PROBLEMS
```

tfactl ips UNPACK FILE

Use the `tfactl ips UNPACK FILE` command to unpack a physical file into a specific path.

Syntax

Running the following command automatically creates a valid `ADR_HOME` structure. The path must exist and be writable.

```
tfactl ips UNPACK FILE file_spec [INTO path]
```

Example

```
$ tfactl ips unpack file /tmp/IPSPKG_20061026010203_COM_1.zip into /tmp/newadr
```

tfactl ips UNPACK PACKAGE

Use the `tfactl ips UNPACK PACKAGE` command to unpack physical files in the current directory into a specific path, if they match the package name.

Syntax

Running the following command automatically creates a valid `ADR_HOME` structure. The path must exist and be writable.

```
tfactl ips UNPACK PACKAGE pkg_name [INTO path]
```

Example

```
$ tfactl ips unpack package IPSPKG_20061026010203 into /tmp/newadr
```

tfactl collection

Use the `tfactl collection` command to stop a running Oracle Trace File Analyzer collection.

Syntax

```
tfactl collection [stop collection_id]
```

You can only stop a collection using the `tfactl collection` command. You must provide a collection ID, which you can obtain by running the `tfactl print` command.

tfactl print

Use the `tfactl print` command to print information from the Berkeley database.

Syntax

```
tfactl print [status | config | directories | hosts | actions | repository | cookie]
```

Parameters

Table 44 tfactl print Command Parameters

Parameter	Description
status	Displays the status of Oracle Trace File Analyzer across all nodes in the cluster. Also, displays the Oracle Trace File Analyzer version and the port on which it is running.
config	Displays the current Oracle Trace File Analyzer configuration settings.
directories	Lists all the directories that Oracle Trace File Analyzer scans for trace or log file data. Also, displays the location of the trace directories allocated for the database, Oracle ASM, and instance.
hosts	Lists the hosts that are part of the Oracle Trace File Analyzer cluster, and that can receive cluster-wide commands.
actions	Lists all the actions submitted to Oracle Trace File Analyzer, such as diagnostic collection. By default, <code>tfactl print</code> commands only display actions that are running or that have completed in the last hour.
repository	Displays the current location and amount of used space of the repository directory. Initially, the maximum size of the repository directory is the smaller of either 10 GB or 50% of available file system space. If the maximum size is exceeded or the file system space gets to 1 GB or less, then Oracle Trace File Analyzer suspends operations and closes the repository. Use the <code>tfactl purge</code> command to clear collections from the repository.
cookie	Generates and displays an identification code for use by the <code>tfactl set</code> command.

Example

The `tfactl print config` command returns output similar to the following:

```
$ tfactl print config
```

```
-----  
.  
|                                     |  
|                                     | node1  
|                                     |  
+-----+-----+  
+  
| Configuration Parameter           | Value  
+-----+-----+  
+  
| TFA Version                       | 12.2.1.0.0  
| Java Version                      | 1.8  
| Public IP Network                 | true  
| Automatic Diagnostic Collection   | true  
| Alert Log Scan                    | true  
| Disk Usage Monitor                | true  
| Managelogs Auto Purge             | false  
| Trimming of files during diagcollection | true  
| Inventory Trace level             | 1  
| Collection Trace level            | 1  
| Scan Trace level                  | 1  
| Other Trace level                 | 1  
| Repository current size (MB)      | 5  
| Repository maximum size (MB)     | 10240  
| Max Size of TFA Log (MB)         | 50  
| Max Number of TFA Logs            | 10  
| Max Size of Core File (MB)       | 20  
| Max Collection Size of Core Files (MB) | 200  
| Minimum Free Space to enable Alert Log Scan (MB) | 500  
| Time interval between consecutive Disk Usage Snapshot(minutes) | 60  
| Time interval between consecutive Managelogs Auto Purge(minutes) | 60  
| Logs older than the time period will be auto purged(days[d]|hours[h]) | 30d  
|
```

Automatic Purging	true
Age of Purging Collections (Hours)	12
TFA IPS Pool Size	5

+-----	

In the preceding sample output:

- **Automatic diagnostic collection:** When `ON` (default is `OFF`), if scanning an alert log, then finding specific events in those logs triggers diagnostic collection.
- **Trimming of files during diagcollection:** Determines if Oracle Trace File Analyzer trims large files to contain only data that is within the specified time ranges. When trimming is `OFF`, no trimming of trace files occurs for automatic diagnostic collection.
- **Repository current size in MB:** How much space in the repository is used.
- **Repository maximum size in MB:** The maximum size of storage space in the repository. Initially, the maximum size is set to the smaller of either 10 GB or 50% of free space in the file system.
- **Trace Level:** 1 is the default, and the values 2, 3, and 4 have increasing verbosity. While you can set the trace level dynamically for running the Oracle Trace File Analyzer daemon, increasing the trace level significantly impacts the performance of Oracle Trace File Analyzer. Increase the trace level only at the request of My Oracle Support.
- **Automatic Purging:** Automatic purging of Oracle Trace File Analyzer collections is enabled by default. Oracle Trace File Analyzer collections are purged if their age exceeds the value of `Minimum Age of Collections to Purge`, and the repository space is exhausted.
- **Minimum Age of Collections to Purge (Hours):** The minimum number of hours that Oracle Trace File Analyzer keeps a collection, after which Oracle Trace File Analyzer purges the collection. You can set the number of hours using the `tfactl set minagetopurge=hours` command.
- **Minimum Space free to enable Alert Log Scan (MB):** The space limit, in MB, at which Oracle Trace File Analyzer temporarily suspends alert log scanning until space becomes free. Oracle Trace File Analyzer does not store alert log events if space on the file system used for the metadata database falls below the limit.

tfactl purge

Use the `tfactl purge` command to delete diagnostic collections from the Oracle Trace File Analyzer repository that are older than a specific time.

Syntax

```
tfactl purge -older number[h | d]
```

Example

The following command removes files older than 30 days:

```
$ tfactl purge -older 30d
```

tfactl managelogs

Use the `tfactl managelogs` command to manage Automatic Diagnostic Repository log and trace files.

Syntax

```
tfactl managelogs [-purge [[-older mm|h|d] | [-gi] | [-database all|d1,d2,...]]]
[-show [usage|variation] [[-older nd] | [-gi] | [-database all|d1,d2,...]]]
```

Parameters

Table 45 tfactl managelogs Purge Options

Purge Option	Description
-older	Time period for purging logs.
-gi	Purges Oracle Grid Infrastructure logs (all Automatic Diagnostic Repository homes under GIBASE/diag and crsdata (cvu dirs)).
-database	Purges Oracle database logs (Default is all, else provide a list).
-dryrun	Estimates logs cleared by <code>purge</code> command.

Table 46 tfactl managelogs Show Options

Show Option	Description
-older	Time period for change in log volume.
-gi	Space utilization under GIBASE.
-database	Space utilization for Oracle database logs (Default is all, else provide a list).

Example

```
$ tfactl managelogs -show usage -gi
```

Output from host : node3

```
-----
.-----
----.
|                               Grid Infrastructure
Usage                           |
+-----+
+-----+
| Location                       |
Size                             |
+-----+
+-----+
| /scratch/app/orabase/diag/clients/user_grid/host_1389480572_107/alert   | 8.00
KB |
| /scratch/app/orabase/diag/clients/user_grid/host_1389480572_107/incident | 4.00
KB |
| /scratch/app/orabase/diag/clients/user_grid/host_1389480572_107/trace   | 1.55
MB |
| /scratch/app/orabase/diag/clients/user_grid/host_1389480572_107/cdump   | 4.00
```

KB			
		/scratch/app/orabase/diag/clients/user_oracle/host_1389480572_107/alert	8.00
KB			
		/scratch/app/orabase/diag/clients/user_oracle/host_1389480572_107/incident	4.00
KB			
		/scratch/app/orabase/diag/clients/user_oracle/host_1389480572_107/trace	
712.00 KB			
		/scratch/app/orabase/diag/clients/user_oracle/host_1389480572_107/cdump	4.00
KB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener/alert	
921.39 MB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener/incident	4.00
KB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener/trace	
519.20 MB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener/cdump	4.00
KB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener_scan2/alert	
726.55 MB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener_scan2/incident	4.00
KB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener_scan2/trace	
339.90 MB			
		/scratch/app/orabase/diag/tnslsnr/node3/listener_scan2/cdump	4.00
KB			
		/scratch/app/orabase/diag/diagtool/user_grid/adrci_1389480572_107/alert	8.00
KB			
		/scratch/app/orabase/diag/diagtool/user_grid/adrci_1389480572_107/incident	4.00
KB			
		/scratch/app/orabase/diag/diagtool/user_grid/adrci_1389480572_107/trace	12.00
KB			
		/scratch/app/orabase/diag/diagtool/user_grid/adrci_1389480572_107/cdump	4.00
KB			
		/scratch/app/orabase/diag/diagtool/user_grid/adrci_1389480572_107/hm	4.00
KB			
		/scratch/app/orabase/diag/crs/node3/crs/alert	44.00
KB			
		/scratch/app/orabase/diag/crs/node3/crs/incident	4.00
KB			
		/scratch/app/orabase/diag/crs/node3/crs/trace	1.67
GB			
		/scratch/app/orabase/diag/crs/node3/crs/cdump	4.00
KB			
		/scratch/app/orabase/diag/asmtool/user_grid/host_1389480572_107/alert	8.00
KB			
		/scratch/app/orabase/diag/asmtool/user_grid/host_1389480572_107/incident	4.00
KB			
		/scratch/app/orabase/diag/asmtool/user_grid/host_1389480572_107/trace	8.00
KB			
		/scratch/app/orabase/diag/asmtool/user_grid/host_1389480572_107/cdump	4.00
KB			
		/scratch/app/orabase/diag/asmtool/user_root/host_1389480572_107/alert	20.00
KB			
		/scratch/app/orabase/diag/asmtool/user_root/host_1389480572_107/incident	4.00
KB			
		/scratch/app/orabase/diag/asmtool/user_root/host_1389480572_107/trace	8.00
KB			
		/scratch/app/orabase/diag/asmtool/user_root/host_1389480572_107/cdump	4.00
KB			

		+-----+	

```
| Total | 4.12
GB |
```

```
'-----'
+-----'
```

```
$ tfactl managelogs -show variation -older 2h -gi
```

```
Output from host : node1
```

```
-----
2016-09-30 00:49:57: INFO Checking space variation for 2 hours
```

```
-----
-----
|                                     Grid Infrastructure
Variation |
+-----+
+-----+-----+
| Directory | Old
Size | New Size |
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan2/trace | 12.00
KB | 12.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan2/incident | 4.00
KB | 4.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_root/host_1342558790_107/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan3/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/crs/node1/crs/alert |
328.00 KB | 404.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_grid/host_1342558790_107/incident | 4.00
KB | 4.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan2/alert | 16.00
KB | 16.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_root/host_1342558790_107/trace | 8.00
KB | 8.00 KB |
+-----+-----+
| /scratch/app/orabase/diag/crs/node1/crs/incident | 4.00
KB | 4.00 KB |
+-----+-----+
```

```

+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan3/incident      | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_root/host_1342558790_107/incident | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan1/alert        | 12.00
KB | 12.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/clients/user_grid/host_1342558790_107/trace | 1.95
MB | 2.42 MB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan3/alert        |
562.34 MB | 726.93 MB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan1/incident      | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener/incident           | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/crs/node1/crs/cdump                       | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener/trace              |
307.22 MB | 394.32 MB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_grid/host_1342558790_107/trace | 12.00
KB | 12.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/clients/user_grid/host_1342558790_107/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/clients/user_grid/host_1342558790_107/incident | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/clients/user_oracle/host_1342558790_107/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/asmtool/user_grid/host_1342558790_107/cdump | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+
| /scratch/app/orabase/diag/clients/user_oracle/host_1342558790_107/incident | 4.00
KB | 4.00 KB |
+-----+-----+
+-----+-----+

```

```

| /scratch/app/orabase/diag/tnslnr/node1/listener_scan1/trace      | 8.00
KB | 8.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan1/cdump    | 4.00
KB | 4.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan3/trace    |
263.64 MB | 340.29 MB |
+-----+
+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener/alert          |
586.36 MB | 752.10 MB |
+-----+
+-----+
| /scratch/app/orabase/diag/clients/user_oracle/host_1342558790_107/trace | 1.17
MB | 1.17 MB |
+-----+
+-----+
| /scratch/app/orabase/diag/clients/user_grid/host_1342558790_107/alert | 16.00
KB | 16.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/clients/user_oracle/host_1342558790_107/alert | 8.00
KB | 8.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/crs/node1/crs/trace                  | 1.63
GB | 1.84 GB |
+-----+
+-----+
| /scratch/app/orabase/diag/asmtool/user_grid/host_1342558790_107/alert | 12.00
KB | 12.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/asmtool/user_root/host_1342558790_107/alert | 12.00
KB | 20.00 KB |
+-----+
+-----+
| /scratch/app/orabase/diag/tnslnr/node1/listener_scan2/cdump    | 4.00
KB | 4.00 KB |
+-----+
+-----+

```

Glossary

action

A database session parameter that is set by an application to identify the action associated with a database request.

affinity

The word 'affinity' is used to describe any strategy that is expected to increase the probability that a work request finds the required data cached in the instance to which the work request is routed.

aggregation

Aggregation is the process of taking a collection of measurements, and combining them to produce an aggregate measure. For example, counting all the work requests that are completed by a given server in a given Performance Class is a form of aggregation. Totaling the CPU time used by all the work requests in a given Performance Class handled by a particular server during a time interval is another form of aggregation.

application

An application is software that runs on a system and provides one or more services to a user or a group of users. Oracle Talent Management, Oracle ERP, Oracle Asset Lifecycle Management, PeopleSoft Campus Solutions, and Oracle Manufacturing are all examples of applications.

An application usually consists of multiple components; there might be a database component, a J2EE component, a client PC component, a batch component, a web component, a Web Services component, and so on.

Automatic Provisioning

Automatic Provisioning attempts to automate, as much as possible, the activities involved in re-tasking a piece of hardware. For example, taking a piece of hardware that has been running with one operating system and one set of application components, and re-deploying the hardware with a different operating system and a different set of application components.

average response time

The average of the response times for all work requests for a Performance Class for a given time period, specified in seconds.

bottleneck

A component or resource that limits the performance of an application or an entire system.

capacity planning

Capacity planning is the act of determining the amount and type of hardware needed to service projected peak user loads. Capacity planning is often done as part of a larger capital equipment budgeting cycle, and usually involves making load projections months into the future.

classifiers

Value matching rules that are applied to attributes of the work request to map work requests to Performance Classes.

closed workload

The amount of work performed in a system in which a fixed number of users interact with the application and each of these users issues a succession of requests. A new request from a user is triggered only after the completion of a previous request by the same user. A user submits a request, waits for the response of that request, thinks for a certain time and then sends a new request. The average time elapsed between the response from a previous request and the submission of a new request by the same user is called the "think time".

A closed workload is also referred to as a session-based workload.

clusterware

Any software that enables groups, or clusters, of connected computers to operate or be controlled as a unit.

conditioned data

Conditioned data is created from raw data in a post-processing step of some kind. Taking averages, removing outliers, filtering, and parameter estimation procedures are all examples of the kind of post-processing that can be used to create conditioned data from raw data.

database services

A database service is a user-created service that is managed by Oracle Clusterware and serves as a database session connection point. A database service can be offered on one or more Oracle RAC instances, and managed on a for-instance basis (for starting and stopping the service).

demand

Demand is a measure of the amount of work being presented to the system and is usually measured in work requests or requests per second.

elapsed time

An elapsed time measurement (also known as a *wall clock time measurement*) is a measurement of a single, contiguous time interval. The elapsed time interval begins at

some time t_1 and ends at another time t_2 , where both times are read from the same clock. The elapsed time measurement is the value of $(t_2 - t_1)$.

end-to-end response time

The expression *end-to-end response time* includes all time spent and all work done from the time a user request is received (for example, from clicking the Submit button in a browser), until the response is sent back to the user in its entirety. End-to-end response time includes time spent in application servers, Oracle Database, Oracle Automatic Storage Management, and traversing the internal networks of the data center.

entry point

The entry point is the initial point of contact between a work request and the Oracle Database QoS Management system. Work requests are initially classified and tagged at their entry point.

fair share scheduling

Fair share scheduling attempts to fairly allocate a resource such as a CPU among a collection of users, ensuring that each user gets a specified share of the available resource. Lottery based scheduling is one kind of fair share scheduling.

Free pool

A server pool that contains servers that are not assigned to any other server pool.

headroom

When a Performance Class is meeting its Performance Objectives, headroom refers to the difference between the actual response times and the required response times, or the surplus in performance.

layer

Layer and tier are synonymous.

Layer Active Time

Layer Active Time is the cumulative time that a work request is actively doing work at a layer, excluding time spent waiting on layers below. Layer Active Time includes time spent executing at the layer, and time spent waiting for layer local resources, such as the CPU, locally connected disks, memory, and so on.

Layer Response Time

Layer Response Time is the elapsed time for a work request to be completely handled by a specific layer. The layer response time includes the time spent executing the work request, and the time spent waiting for local and remote resources and servers.

layer visit

Often, a single work request from an end user (for example, clicking a link in a browser) causes several requests to arrive at various layers of the system. Each time a request is handled by a layer is called a layer visit.

load shedding

Load shedding refers to the act of rejecting some work requests, so that other work requests can complete successfully. Rejecting requests gracefully might require modifications to your applications. For example, you might want the end user to see a customized rejection page. Alternatively, you might want to store information from the work request so you can reply to the requester at a later time.

lottery based scheduling

Lottery based scheduling is a scheduling algorithm that uses random numbers to apportion resources (such as a CPU) among a collection of users, according to a pre-set distribution.

memory pressure

A state indicating that there is a limited amount of available memory on a server.

metric

A metric is something that can be measured.

module

Module is the database session parameter that is set by an application, generally to identify the application module making the database request.

open workload

Work performed in a system in which new work requests to an application come from outside the system being managed. The work requests are independent of each other and the work request arrival rate is not influenced by the response time for previous requests, or the number of requests that have already arrived and are being processed. The number of work requests that the system might be asked to execute at any given time can range from zero to infinity. The system's resources or servers perform various activities to process a work request and the work request leaves the system when processing is complete.

Open workloads are also referred to as request-based workloads.

Oracle Grid Infrastructure for a cluster

A term assigned to the software stack comprising Oracle's generic Clusterware, Oracle Automatic Storage Management (Oracle ASM), Oracle RAC agents, and the Oracle RAC database management infrastructure layer.

Oracle Database Resource Manager

Oracle Database Resource Manager is a software component available with the Oracle Database. Oracle Database Resource Manager enables an administrator to establish Resource Plans that control how various resources (such as the CPU) are allocated to consumer groups, which are collections of work requests. The intent is very similar to Oracle Database QoS Management's Performance Class.

performance bottleneck

Oracle Database QoS Management attempts to identify performance bottlenecks due to Performance Classes waiting too long for required resources, such as CPU, Global Cache, or I/O.

Performance Class

A Performance Class is a group of related work requests. Performance Objectives are written for a Performance Class. All work requests that are grouped into a particular Performance Class have the same performance objective.

Performance Class ranks

The Performance Class rank represents the business criticalness of each Performance Class in a set of Performance Objectives that are in effect at a given time. When there are not enough resources available to service all applicable Performance Classes at the same time, Oracle Database QoS Management works to meet the Performance Objectives for the Performance Classes that are highest ranked at the expense of Performance Classes with a lesser rank. For example, Performance Classes with an rank of Lowest are sacrificed if necessary to ensure that Performance Classes of higher rank (Highest, High, Medium and Low) continue to meet their Performance Objectives.

performance objectives

Performance objectives refers to business level objectives for the system. A performance objective includes both Performance Objectives and availability objectives.

Performance Objectives

A Performance Objective defines a level of performance that is optimal for business purposes for a given Performance Class. For a particular Performance Class, a Performance Objective specifies the target average response time for that workload.

In high load situations, work of lower business criticalness can be deliberately starved for resources by the Oracle Database QoS Management system so that more important work can meet its Performance Objectives; in this circumstance the user might receive a "Server Busy" message instead of just experiencing very poor response times.

Performance Policy

A Performance Policy is a collection of Performance Objectives and Performance Class ranks that are intended to be in force at the same time. A Performance Policy must include at least one Performance Objective and Performance Class rank for each Performance Class, unless the Performance Class is marked Measure-Only. A Performance Policy optionally includes server pool directive overrides to set a baseline configuration of server resources for the time period in which the policy is active.

Performance Satisfaction Metric

A normalized numeric value that indicates how well a particular Performance Objective is being met, and which enables Oracle Database QoS Management to compare the performance of the system for widely differing Performance Objectives.

Policy Set

A Policy Set is a wizard-generated XML document that governs the operation of Oracle Database QoS Management. A Policy Set specifies server pools and their hosted Performance Classes, the collection of Performance Policies that specify the Performance Objectives for each Performance Class, and the server pool directive overrides for each Performance Policy.

program name

Program name is a database session attribute set by an application that is generally used to identify the program making the database request.

raw data

Raw data is data that has not been post-processed in any way. Counts, totals, and individual sample values are examples of raw data.

resource

A resource is a shared item that has limited quantity that is required to process a request. For example, CPU Time, threads, memory, I/O devices, slots in queues, network bandwidth, and temp space are all resources. Servers typically provide resources.

resource allocation control

A resource allocation control (also informally known as a *knob*) is a parameter, or collection of parameters, to a resource allocation mechanism. Examples of a resource allocation control include:

- A Consumer Group for Oracle Database Resource Manager
- The number of servers in a server pool

resource allocation mechanism

A resource allocation mechanism is something that gives an external entity such as a person or Oracle Database QoS Management the ability to control how some collection of resources are allocated. Oracle Database Resource Manager is an example of a Resource Allocation Mechanism.

resource metric

A resource metric is a metric that can be measured for any resource. Examples include Resource Usage Time and Resource Wait Time.

Resource Usage Time

Resource usage time is the cumulative time that a work request had exclusive use of a resource.

resource use

Resource use is a measurement that accumulates a specified set of elapsed time measurements into a single number. For example, a measurement of the CPU time spent on a given work request on a given server is a resource measurement: the specified work request uses the CPU for many separate intervals of time as the work request is processed.

resource wait time

Resource wait time is the cumulative time spent waiting for a resource by a work request that is ready to use that resource.

response time

The time between the server receiving a transaction request and sending out a response after committing or aborting the transaction.

rogue work

A work request that uses significantly more resources than expected; for example, the work request might be in a non-terminating loop. In some systems, facilities are provided to stop or re-prioritize rogue work.

routing

Routing is the act of choosing the path that a work request takes through the system. This includes all choices made when picking an entity in another tier of the system to which to pass a work request.

server

A server is a shared computer, typically not dedicated to a single user. A server can be as simple as a single CPU blade, or as complex as a large server with many CPUs sharing memory.

server pools

A server pool is a collection of servers created by the cluster administrator using either Enterprise Manager Cloud Control or the Server Control (SRVCTL) utility. Server pools are contained within tiers; each service is assigned to run in a specific server pool.

server pool directive overrides

High availability guidelines for the cluster administrator server to keep the cluster highly available.

server pool importance

A number from 0 to 1000 (0 being least important) that ranks a server pool among all other server pools in a cluster.

server pool maximum

The maximum number of servers that the server pool should contain.

server pool minimum

The minimum number of servers that the server pool should contain.

service

A service provides a well-recognized value to a user (client) or group of users. A service is provided to an application, and runs on a system. For example, CollabSuite provides a set of services such as Files, Calendar, Web Conferences, and so on. See also [database services](#).

Operational management decisions, such as the hours of operation, capacity planning, provisioning, placement, and so on, are made on a service-by-service basis.

service placement

The activities of starting, stopping, or relocating a database service

singleton services

Services within a server pool that has a size of one.

system

A shared collection of servers and their associated infrastructure (networks, firewalls, storage systems, and so on) over which a workload management instance operates.

system metric

System metrics are metrics that help us to connect the things that are happening at the different layers. They provide a framework within which the rest of the analysis can be done. Examples include request counts, Layer Response Time, Layer Active Time, and so on.

All tiers of the system must provide the same set of system metrics.

tag

When a work request is received by the system, an attempt is made to classify the type of work requested. The objective of classification is to determine which Performance Objective applies to this particular work request. The result of classification is a tag (the Performance Class name) that is carried with the work request as it proceeds through the system. The tag enables the work request to be associated with the Performance Objective for the workload (Performance Class).

tier

A tier is a logical processing component within a system. The tiers are stacked on top of each other to provide the end-to-end processing stack for a system. WebCache, OHS, OC4J, Oracle Database and Oracle Automatic Storage Management are examples of tiers.

There can be multiple entities in a given tier providing either redundancy or distinct functionality. For example, a system might include two OHS instances for higher availability and two databases, one for CRM, and the other for ERP.

uniform services

Services that must be offered on every node of a server pool.

UserName

The OCI_ATTR_USERNAME or the Oracle Database user that is used to authenticate to the database.

work request

A work request is the smallest atom of work that a user can initiate. A work request can be an HTTP request, a SOAP request, a SQL statement sent to the database, or the execution of a process. A work request arrives at a layer, perhaps from the outside world, perhaps from another layer. The work request is processed, and a response is generated; the response is sent back to the requester.

Index

A

ACL attributes, [10-3](#)
ACTION, [6-12](#), [7-5](#)
activating
 Performance Policies, [9-26](#)
active resource plan, [10-2](#)
adding
 classifiers to a Performance Class, [9-21](#)
 servers to server pools, [10-4](#)
 services to the Policy Set, [9-16](#), [9-17](#)
administrator-managed databases
 support for, [6-10](#)
administrators
 cluster, [8-2](#), [8-5](#), [10-3](#)
 creating for Oracle Database QoS Management, [9-33](#)
 creating the QoSAdmin user, [8-5](#)
 database, [10-3](#)
 initial configuration tasks, [8-1](#)
 Oracle Database QoS Management, [8-5](#)
alerts
 in Oracle Enterprise Manager, [6-20](#), [7-11](#)
alternative recommendations, [6-22](#), [9-9](#)
application contexts, [7-5](#)
applications
 classifying work requests, [6-12](#)
 definition, [6-26](#)
 parameters used by classifiers, [6-12](#)
 resource requirements, [6-7](#)
 supported connections, [7-4](#)
APPQOS_PLAN, [8-7](#), [10-2](#)
 editing, [9-35](#)
APPQOSSYS user, [8-7](#), [10-2](#)
audit log, [7-11](#)
authorized actions, [9-22](#), [9-27](#)
automated risk identification, [1-4](#)
Automatic Database Diagnostic Monitor (ADDM), [7-7](#)
automatic diagnostic collections, [15-1](#)
Automatic Diagnostic Repository
 log file, [21-1](#)
 trace file, [21-1](#)
Automatic Diagnostic Repository (ADR), [12-1](#), [12-4](#)

automatic purging, [19-4](#)
automatic restart, [2-2](#)
Automatic Workload Repository (AWR), [7-7](#)
AUTORUN_FLAGS
 exclude profile, [2-20](#)
 profile, [2-20](#)
 tag, [2-20](#)
AUTORUN_SCHEDULE, [8](#)
availability issues
 database, [1-1](#)
 server, [1-1](#)
average response times
 and Performance Classes, [7-10](#)
 and Performance Objectives, [6-14](#), [7-4](#)
 definition, [6-14](#), [6-31](#)

B

batch jobs, [6-15](#)
bequeath connections, [7-4](#)
bottlenecks
 and Performance Classes, [7-6](#), [9-32](#)
 and Performance Objectives, [6-32](#)
 and recommendations, [7-11](#)
 CPU resource, [7-6](#)
 for Performance Classes, [6-32](#)
 Global Cache resource, [7-7](#)
 I/O resource, [7-7](#)
 locating using metrics, [6-32](#)
 Other resource, [7-7](#)
 resolving, [6-22](#)
business objectives, [6-7](#)

C

CA-signed certificate, [19-10](#)
calendars, [6-6](#)
candidate server lists, [8-2](#)
capture debug output, [2-54](#)
CDB, [6-22](#), [6-28](#)
certificates, [22-1](#)
CGroups, [8-8](#)
CHACTL, [3-3](#), [3-9](#)
 commands
 chactl calibrate, [38](#)

- CHACTL (*continued*)
 - commands (*continued*)
 - chactl config, 38
 - chactl export, 46
 - chactl import, 47
 - chactl monitor, 35
 - chactl query calibration, 43
 - chactl query diagnosis, 40
 - chactl query model, 42
 - chactl query repository, 43
 - chactl remove, 46
 - chactl rename, 46
 - chactl resize repository, 48
 - chactl set maxretention, 47
 - chactl status, 37
 - chactl unmonitor, 36
- chmod, 2-56
- classification rules
 - defaults, 6-12
- classifiers, 6-11, 6-12, 7-5
 - adding, 9-21
 - and multiple services, 7-5, 9-18
 - and Performance Classes, 7-5, 9-19
 - and service names, 6-12
 - and session attributes, 7-5
 - and tags, 7-5
 - creating, 9-18
 - default_pc, 9-22
 - deleting, 9-21
 - evaluation order, 10-4
 - for the default Performance Class, 9-22
 - modifying, 9-21
 - multiple attributes, 7-5
 - order of evaluation, 7-5
 - session attributes, 7-5
 - specifying evaluation order, 9-12, 9-22
- clients
 - Java Database Connectivity (JDBC) clients, 7-4
 - name of program, 7-5
 - Oracle Call Interface (OCI), 7-4
 - supported connections, 7-4
 - tracking across database sessions, 7-5
- cluster administrator, 8-2, 8-5, 9-33, 10-3
- Cluster Health Monitor, 4-1
 - collecting Cluster Health Monitor data, 4-2
 - data
 - historical, 4-2
 - real time, 4-2
 - node eviction, 1-5
 - OCLUMON, 10
 - pinned process, 1-5
- Cluster Health Monitor services
 - cluster logger, 4-1
- Cluster Health Monitor (*continued*)
 - services (*continued*)
 - system monitor, 4-1
 - cluster or host view, 20-2
 - cluster resource activity log, 5-3, 25
 - cluster resource failures
 - monitoring, 5-3
 - cluster_database, 12-1
 - collection period, 18-1
 - command interfaces, 14-5
 - configuration recommendations, 7-6
 - configuration violations, 7-1, 7-2
 - configuring
 - load balancing goals, 7-3
 - Oracle Grid Infrastructure for a cluster, 8-1
 - Oracle RAC databases, 8-1
 - Performance Objectives, 9-22
 - server pool directive overrides, 9-15, 9-24, 9-27
 - server pools, 7-2
 - connecting to a database, 7-3
 - connection load balancing, 7-3
 - connection pools, 7-3
 - connections
 - bequeath, 7-4
 - load balancing, 7-3
 - load balancing at run time, 7-3
 - to databases, 7-4
 - consumer groups, 6-4, 6-26
 - recommendations, 6-20, 6-21
 - container database, 6-22
 - container database (CDB), 7-2
 - control groups, 8-8
 - copying Performance Policies, 9-25
 - CPU partition, 6-4
 - CPU resource bottlenecks
 - resolving, 7-6
 - CPU shares, 6-22
 - CPU_COUNT, 6-4, 10-3
 - and multiple databases, 7-2, 8-4
 - default value, 7-2, 8-4
 - CPUS
 - node view, 12
 - create incident tickets, 2-45
 - Create Policy Set wizard, 8-7
 - creating
 - initial Policy Set, 8-7
 - Oracle Database QoS Management administrator user, 9-33
 - Performance Classes, 9-18
 - Performance Policies, 9-22
 - QoSAdmin user, 8-5
 - server pools, 8-2, 9-15
 - services, 8-4
- CRSCTL, 10-3

CRSCTL (*continued*)

- commands
 - get calog maxsize, [32](#)
 - get calog retentiontime, [32](#)
 - query calog, [25](#)
 - set calog maxsize, [33](#)
 - set calog retentiontime, [33](#)

csv format file output, [12](#)

custom application integration, [2-51](#)

D

daemon

- force stop, [2-17](#)
- info, [2-26](#)
- initcheck, [2-18](#)
- initrmsetup, [2-18](#)
- initsetup, [2-18](#)
- nextautorun, [2-26](#)
- passwordless SSH, [2-18](#)
- start, [2-17](#)
- status, [2-26](#)
- stop, [2-17](#)

daemon mode operation, [2-2](#)

Dashboard, [6-20](#), [9-4](#)

- accessing, [9-4](#)
- analysis result for recommendations, [9-8](#)
- and recommendations, [6-20](#)
- and resource bottlenecks, [7-6](#)
- and workload surges, [7-11](#)
- description of General section, [9-4](#)
- implementing recommendations, [9-11](#)
- logging in, [8-5](#)
- main sections, [9-4](#)
- Performance Overview, [9-4](#), [9-7](#)
- Performance Satisfaction Metric chart, [9-7](#)
- QoS Status, [9-4](#)
- recommendations not generated, [10-4](#)
- Recommendations section, [9-4](#)
- Resource Use vs. Wait Time chart, [9-7](#)
- Resource Wait Times Breakdown, [9-4](#), [9-32](#)
- Situation Analysis, [9-9](#)
- viewing recommendation details, [9-9](#)
- viewing recommendations, [9-8](#)

data redaction, [14-5](#)

database

- startup order determined by rank, [6-18](#)

database alert logs, [12-4](#)

Database Configuration Assistant (DBCA), [8-2](#)

database connections, [6-5](#)

database parameters, [6-29](#)

Database Resource Manager

See Oracle Database Resource Manager

database services

See services

database sessions

- tracking client requests, [7-5](#)

databases

- adding services, [9-16](#), [9-17](#)
- and CPU_COUNT, [7-2](#), [8-4](#)
- compliant, [10-2](#)
- connecting with a service, [7-3](#)
- connections, [7-4](#)
- default service, [7-3](#)
- enabling Oracle Database QoS Management, [10-2](#)
- Oracle RAC One Node, [7-2](#)
- policy-managed, [6-10](#), [7-2](#), [10-2](#)
- services, [7-2](#)
- sharing a server pool, [7-2](#)
- singleton, [7-2](#)
- upgrading, [6-10](#)

defaults

- and classifiers, [9-22](#)
- and CPU_COUNT, [7-2](#), [8-4](#)
- database services, [7-3](#)
- namespace, [7-5](#)
- password for QoSAdmin user, [9-34](#)
- Performance Policy, [7-10](#), [9-12](#)
- Policy Set, [6-6](#), [7-10](#)
- QoSAdmin password, [8-5](#)

deleting

- classifiers for a Performance Class, [9-21](#)
- Performance Classes, [9-19](#)
- Performance Policies, [9-27](#)

demand

- changing levels of, [6-7](#)
- surges, [6-1](#), [6-3](#), [6-7](#), [6-19](#)

detailed view, [20-3](#)

DEVICES

- node view, [12](#)

diff report, [2-16](#)

disabling

- Oracle Database QoS Management for a cluster, [9-6](#)

E

edit incident tickets, [2-46](#)

Edit Policy Set wizard, [9-11](#), [9-12](#), [9-18–9-20](#)

Elasticsearch, [2-50](#)

email notification, [2-3](#), [15-2](#)

enabling

- Oracle Database QoS Management, [8-5](#)
- Oracle Database QoS Management for a cluster, [7-10](#), [8-8](#), [9-6](#), [10-1](#)
- Oracle Database QoS Management for a database, [10-2](#)

tracing, [10-5](#)

Enterprise Manager

Enterprise Manager (*continued*)
 integration with Oracle Database QoS
 Management server, [6-18](#)
 evaluating classifiers, [9-19](#)
 evaluation order of classifiers, [9-22](#)
 Expect utility, [2-12](#), [22-1](#)

F

fast application notification (FAN), [7-3](#)
 FAST_START_MTTR_TARGET, [6-29](#)
 file attribute check, [2-29](#)
 file browser, [20-4](#)
 file viewer, [20-5](#)
 FILESYSTEMS
 node view, [12](#)
 floaters, [7-10](#)
 Free pool, [6-10](#), [8-2](#)

G

Global Cache resource bottlenecks
 resolving, [7-7](#)
 goals
 and services, [7-3](#)
 LONG, [7-3](#)
 SEVICE_TIME, [7-3](#)
 SHORT, [7-3](#)
 THROUGHPUT, [7-3](#)
 Grid Infrastructure Management Repository
 (GIMR), [4-1](#)
 GV\$ view queries, [7-4](#)

H

Hang Manager, [1-8](#), [12-1](#)
 headroom, [9-7](#)
 maximizing, [7-10](#)
 health check score and summary, [2-3](#)
 Health Prognostics Engine, [3-2](#)
 high availability, [6-29](#)
 HTML output, [3-4](#)

I

I/O resource bottlenecks
 resolving, [7-7](#)
 implementing
 alternative recommendations, [9-9](#)
 recommendations, [9-11](#)
 import CA certificates, [20-6](#)
 init.tfa, [22-2](#)
 initial Policy Set, [8-7](#)
 initialization parameters

initialization parameters (*continued*)
 CPU_COUNT, [7-2](#), [8-4](#)
 install
 Linux, [14-2](#), [14-3](#)
 Microsoft Windows, [14-3](#)
 UNIX, [14-2](#), [14-3](#)
 instance caging, [6-4](#), [8-8](#)
 integration, [2-48](#)
 Interconnects page
 monitoring Oracle Clusterware with Oracle
 Enterprise Manager, [5-1](#)
 investigate logs, [16-2](#)
 IPS packages, [18-6](#)

J

Java Database Connectivity, [6-5](#), [6-12](#)
 clients, [7-4](#)
 Java exception, [22-12](#)
 Java keytool, [19-9](#), [19-10](#)
 JDBC
 See Java Database Connectivity
 JSON output results, [2-50](#)

K

key directories, [14-4](#)
 Kibana, [2-50](#)
 KPISET parameters, [3-6](#)

L

LMS processes, [6-18](#)
 load balancing, [7-3](#)
 configuring goals for a service, [7-3](#)
 connections, [7-3](#)
 run-time connection, [7-3](#)
 load balancing advisory, [7-3](#)
 data, [7-3](#)
 goal, [6-5](#), [7-3](#)
 lockcells, [2-47](#)
 log files, [10-5](#)

M

maintenance windows, [6-6](#)
 manage diagnostic collections, [19-6](#)
 manage directories, [19-5](#)
 managed server pools
 and Oracle RAC database instances, [7-2](#)
 managing
 server pools, [8-7](#), [10-3](#)
 users, [9-33](#)
 manual purging, [19-5](#)

Maximum Availability Architecture (MAA)
Scorecard, [2-3](#)

MCB

See Multi-CPU Binding

measure-only, [9-22](#)

Performance Classes, [6-14](#), [9-22](#), [9-24](#)

Performance Policies, [7-10](#)

measure-only mode, [7-10](#)

Memory Guard, [1-7](#), [11-1](#)

log file, [11-2](#)

merged files, [20-5](#)

metrics, [6-30](#)

collecting, [6-18](#)

for performance, [6-31](#)

for Performance Classes, [9-4](#)

for resources, [6-31](#)

not displayed, [10-4](#)

performance satisfaction, [6-32](#)

metrics view, [20-2](#)

minimum response times, [7-10](#)

minimum size of server pools, [7-10](#)

modifying

classifiers for a Performance Class, [9-21](#)

Performance Classes, [9-20](#)

Performance Policies, [9-24](#)

server pool settings, [9-15](#)

MODULE, [6-12](#), [7-5](#)

Multi-CPU Binding, [8-8](#)

multi-tier environments, [7-5](#)

multitenant database, [6-28](#)

multitenant databases, [7-2](#)

N

namespaces

default, [7-5](#)

NICS

node view, [12](#)

node view

defined, [12](#)

node views

CPUS, [12](#)

DEVICES, [12](#)

FILESYSTEMS, [12](#)

NICS, [12](#)

PROCESSES, [12](#)

PROTOCOL ERRORS, [12](#)

SYSTEM, [12](#)

TOP CONSUMERS, [12](#)

non-root access, [22-3](#)

non-root users, [22-13](#)

NOTIFICATION_EMAIL, [2-3](#), [8](#)

O

OC4J container

locating, [8-5](#)

OCHAD daemon, [3-2](#)

OCI

See Oracle Call Interface

OCIAttrSet function, [7-5](#)

OCLUMON

commands

debug, [11](#)

dumpnodeview, [12](#)

manage, [23](#)

version, [24](#)

ODP.NET, [6-12](#)

OLTP workloads, [7-4](#)

on-demand diagnostic collection, [16-1](#)

on-demand mode, [2-12](#)

open workloads, [6-1](#)

openssl, [19-10](#)

ORA\$QOS_CDB_PLAN, [10-2](#)

editing, [9-35](#)

ORA\$QOS_PLAN

editing, [9-35](#)

Oracle Call Interface, [6-5](#), [6-12](#)

clients, [7-4](#)

Oracle Cluster Health Advisor, [1-6](#), [3-1](#)

Oracle Cluster Health Advisor daemon, [3-2](#)

Oracle Cluster Health Advisor model, [3-8](#)

Oracle Clusterware

monitoring resources, [5-1](#)

monitoring with Oracle Enterprise Manager,
[5-1](#)

resources

monitoring, [5-1](#)

Oracle Clusterware Repository, [8-7](#), [9-33](#)

Oracle Database QoS Management, [1](#)

administrator, [8-1](#), [8-5](#)

changing active Performance Policy, [9-26](#)

Dashboard, [9-4](#)

See also Dashboard

Dashboard sections, [9-4](#)

demand

surges, [1-8](#)

disabling for a cluster, [9-6](#)

enabling for a cluster, [7-10](#), [8-5](#), [8-8](#), [9-6](#)

enabling for a database, [8-5](#)

log files, [10-5](#)

open workloads, [1-8](#)

selecting server pools, [9-12](#)

server hosting the OC4J container, [8-5](#)

status, [9-4](#)

trace files, [10-5](#)

using the Dashboard, [9-4](#)

viewing cluster management status, [9-2](#)

- Oracle Database QoS Management (*continued*)
 - viewing current status, [8-8](#)
 - work requests, [1-8](#)
- Oracle Database QoS Management
 - administrator, [9-4](#)
 - creating, [8-5](#)
 - default password, [8-5](#)
- Oracle Database Resource Manager, [6-4](#), [6-26](#), [9-35](#)
 - enabling, [10-2](#)
- Oracle Enterprise Manager, [2-48](#)
 - using the Interconnects page to monitor
 - Oracle Clusterware, [5-1](#)
- Oracle Enterprise Manager alerts, [7-11](#)
- Oracle Enterprise Manager Dashboard, [6-20](#)
 - See also Dashboard
- Oracle Health Check Collections Manager, [2-32](#)
 - bulk mapping systems to business units, [2-35](#)
 - email notification system, [2-8](#)
 - failed uploads, [2-40](#)
 - incident tab, [2-45](#)
 - incident tracking system, [2-27](#)
 - purge old collections, [2-36](#)
 - selectively capture users during logon, [2-33](#)
 - upload collections automatically, [2-38](#)
 - user-defined checks, [2-41](#)
- Oracle ORAchk and EXAchk command-line
 - options, [2-28](#), [1](#)
 - daemon options, [8](#)
 - file attribute changes, [2-29](#)
 - baseline snapshot, [2-31](#)
 - exclude directories, [2-30](#)
 - file attribute snapshots, [2-29](#)
 - include directories, [2-30](#)
 - recheck changes, [2-31](#)
 - remove snapshots, [2-32](#)
 - restrict system checks, [2-32](#)
 - snapshots, [9](#)
 - generic commands, [4](#)
 - managing the report output, [6](#)
 - scope of checks, [5](#)
 - uploading results to database, [7](#)
- Oracle ORAchk and Oracle EXAchk, [1-4](#), [2-1](#), [2-2](#)
 - AUTORUN_FLAGS, [2-20](#)
 - AUTORUN_SCHEDULE, [2-19](#)
 - collection_retention, [2-21](#)
 - daemon, [2-15](#)
 - get, [2-23](#)
 - nodaemon, [2-15](#)
 - NOTIFICATION_EMAIL, [2-20](#), [2-21](#)
 - PASSWORD_CHECK_INTERVAL, [2-21](#)
 - sendemail, [2-16](#)
 - set, [2-18](#), [2-22](#)
- Oracle ORAchk and Oracle EXAchk (*continued*)
 - testemail, [2-20](#)
 - troubleshoot, [2-53](#)
- Oracle ORAchk and Oracle EXAchk
 - prerequisites
 - Expect utility, [2-8](#)
 - handling of root passwords, [2-8](#)
 - run as
 - Oracle Database home owner, [2-6](#)
 - Oracle Grid Infrastructure home owner, [2-6](#)
 - root, [2-6](#)
- Oracle RAC, [11-4](#)
- Oracle RAC databases
 - creating, [8-2](#)
 - supported releases, [7-2](#)
- Oracle RAC high availability framework, [7-3](#)
- Oracle RAC One Node, [11-4](#)
- Oracle RAC One Node database, [11-3](#), [12-3](#)
- Oracle RAC One Node databases, [7-2](#)
- Oracle Real Application Clusters (Oracle RAC), [3-1](#), [6-26](#), [11-1](#)
 - and Oracle Database QoS Management, [6-26](#)
- Oracle Trace File Analyzer, [1-6](#), [13-1](#)
 - configuration, [19-1](#)
 - configure hosts, [19-7](#)
 - configure ports, [19-7](#)
 - daemon, [22-14](#)
 - discovery, [22-11](#)
 - managing Oracle Trace File Analyzer, [19-3](#)
 - on-demand diagnostic collections
 - custom collections
 - changing the collection name, [18-4](#)
 - copying zip files, [18-5](#)
 - preventing collecting core files, [18-6](#)
 - silent collection, [18-6](#)
 - specific components, [18-2](#)
 - specific directories, [18-3](#)
 - specific nodes, [18-2](#)
 - trimming files, [18-5](#)
 - purge logs automatically, [21-2](#)
 - restarting, [19-3](#)
 - shutting down, [19-3](#)
 - starting, [19-3](#)
 - status, [19-1](#)
 - stopping, [19-3](#)
 - TFACTL
 - command-line utility, [48](#)
- Oracle Trace File Analyzer log analyzer utility, [58](#)
- Oracle Trace File Analyzer service, [20-1](#)
- Oracle Wallet, [8-7](#)
- OSWatcher, [22-10](#)

oswbb, [22-12](#)
 overloaded systems, [6-22](#)
 overrides
 See server pool directive overrides

P

parallel queries, [7-4](#)
 PDB, [6-28](#)
 PDB shares, [6-28](#)
 performance class
 ranks, [6-18](#)
 Performance Classes, [6-6](#), [6-11](#), [6-12](#), [6-32](#),
 [7-11](#), [9-22](#)
 adding classifiers, [9-21](#)
 adding services, [9-16](#), [9-17](#)
 and alerts, [7-11](#)
 and classifiers, [7-5](#)
 and homogenous requests, [7-4](#)
 and services, [6-12](#)
 changing the rank, [9-24](#)
 classifier ordering, [10-4](#)
 classifiers, [7-5](#)
 configuring Performance Objectives, [9-7](#)
 creating, [9-18](#)
 creating classifiers, [9-18](#)
 default, [9-22](#)
 default names, [6-6](#)
 deleting, [9-19](#)
 deleting classifiers, [9-21](#)
 evaluating classifiers, [9-19](#)
 improvements in performance, [6-22](#)
 Incomplete data, [10-4](#)
 list of, [9-4](#)
 managing work requests separately, [6-13](#)
 maximum, [7-5](#)
 measure-only, [6-14](#), [9-7](#)
 metrics, [9-4](#)
 metrics missing, [10-4](#)
 modifying, [9-20](#)
 modifying classifiers, [9-21](#)
 multiple classifiers, [9-19](#)
 No demand, [10-4](#)
 Nonconforming data, [10-4](#)
 projected impact of recommendations, [6-18](#)
 ranks, [6-17](#)
 relation to Performance Objectives, [6-14](#)
 renaming, [9-20](#)
 satisfying Performance Objectives, [6-18](#)
 setting evaluation order of classifiers, [9-12](#)
 specifying evaluation order of classifiers,
 [9-22](#)
 specifying ranks, [9-22](#)
 suggested by application developer, [6-13](#)
 unable to collect metrics, [10-3](#)

Performance Classes (*continued*)
 used in Performance Policies, [7-10](#)
 viewing performance metrics, [9-7](#), [9-29](#), [9-30](#)
 viewing resource wait times, [9-4](#)
 wait times for resources, [9-32](#)
 performance issues
 database client, [1-3](#)
 database server, [1-3](#)
 performance metrics, [6-31](#)
 performance objectives
 importance, [6-14](#)
 supported types, [7-4](#)
 Performance Objectives, [6-6](#), [6-32](#)
 and services, [7-10](#)
 configuring, [9-7](#)
 definition, [6-14](#)
 managing, [6-14](#)
 ranks, [6-19](#), [7-10](#)
 relation to Performance Classes, [6-14](#)
 specifying, [9-22](#)
 specifying values, [9-22](#)
 Performance Policies, [6-6](#), [6-14](#), [6-19](#)
 activating, [9-26](#)
 and server pools, [7-2](#)
 changing active, [9-26](#)
 copying, [9-25](#)
 creating, [9-22](#)
 creating additional, [7-10](#)
 default, [7-10](#)
 Default policy, [9-12](#)
 definition, [6-14](#)
 deleting, [9-27](#)
 measure-only, [7-10](#)
 modifying, [9-24](#)
 ranking Performance Classes, [6-17](#)
 refining with Performance Classes, [7-10](#)
 scheduling the active policy, [9-26](#)
 server pool directive overrides, [6-15](#)
 performance projections, [7-11](#)
 Performance Satisfaction Metric (PSM), [6-32](#),
 [7-11](#), [9-7](#), [9-22](#)
 performance tuning, [7-7](#)
 Perl, [22-2](#)
 pluggable database (PDB), [7-2](#)
 pluggable databases
 monitoring, [6-22](#)
 Policy and Performance Management engine,
 [6-18](#)
 policy changes
 logs, [7-11](#)
 Policy Set
 active policy, [9-4](#)
 adding server pools, [9-15](#)
 administrating, [9-11](#)
 creating a new Policy Set, [9-11](#)

Policy Set (*continued*)
 current policy, 9-4
 default, 6-6
 initial, 6-6
 modifying, 9-12
 Policy Sets, 6-6
 contents, 6-6
 creating, 8-7
 default, 6-12, 7-10
 default settings, 8-7
 policy-managed databases, 7-2, 10-2
 privileged user
 finding, 2-44
 proactive notification, 1-4
 PROCESSES
 node view, 12
 PROGRAM, 6-12, 7-5
 PROTOCOL ERRORS
 node view, 12

Q

QoS administrator
 initial configuration tasks, 8-1
 qosadmin, 8-5
 QoSAdmin user, 8-5, 9-4
 default password, 8-5, 9-34
 qosctl, 8-5
 QOSCTL utility, 9-33
 syntax, 9-34
 qosmsserver
 default port, 10-5

R

ranking Performance Objectives, 7-10
 ranks
 and Performance Classes, 6-17, 9-24
 Performance Classes, 9-22
 recommendations, 6-6, 6-18, 6-29, 7-11, 9-32
 alternatives, 6-22, 9-9
 analysis results, 9-8
 automatically implemented, 9-27
 choosing a different action, 9-9
 consumer group mappings, 6-20, 6-21
 contents, 6-23
 generating, 6-22
 implementing, 6-20, 6-22, 9-11
 implementing alternative recommendations,
 9-9
 interval, 9-7
 iterations, 6-25
 negative impacts, 6-22
 not being displayed, 9-11
 not generated, 10-4

recommendations (*continued*)
 reallocating servers, 6-21
 section of the Dashboard, 9-4
 Situation Analysis, 9-9
 viewing, 9-4, 9-8
 viewing details, 9-9
 Recommended Actions page, 9-9
 remote login, 2-55
 removing servers from a server pool, 7-2
 renaming
 Performance Classes, 9-20
 report findings, 2-3
 report overview, 2-3
 repository, 22-4
 requirements
 system, 7-1, 7-6
 resolving
 CPU resource bottlenecks, 7-6
 Global Cache resource bottlenecks, 7-7
 I/O resource bottlenecks, 7-7
 Other resource bottlenecks, 7-7
 resource bottlenecks
 resolving, 7-7
 Resource Manager
 See Oracle Database Resource Manager
 resource plans
 APPQOS_PLAN, 8-7, 10-2
 editing, 9-35
 ORA\$QOS_CDB_PLAN, 10-2
 used by Oracle Database QoS Management,
 6-28
 resource types, 7-6
 bottlenecked, 9-32
 CPU, 7-6
 Global Cache, 7-7
 I/O, 7-7
 Other, 7-7
 qosmsserver, 10-5
 viewing wait times, 9-4
 Resource Use vs. Wait Time chart, 9-7
 resources
 allocating, 6-3
 bottlenecks
 and recommendations, 6-18
 increasing CPU access, 6-25
 monitoring, 25
 resolving bottlenecks, 6-22
 starved, 6-22
 types of, 6-31
 usage, 6-31
 wait times, 6-31
 waits, 6-30
 response time, 6-14
 response times
 average per request, 7-4

response times (*continued*)
 definition, [6-30](#)
 minimum achievable, [7-10](#)
 RMI port conflict, [10-5](#)
 run-time connection load balancing, [7-3](#)
 run-time connection load balancing goal, [6-5](#)

S

schedule email health check reports, [1-4](#)
 scheduling
 active performance policy, [9-26](#)
 scripting, [9-26](#)
 self-signed certificate, [19-9](#)
 sensitivity, [12-3](#)
 server pool directive overrides, [6-15](#), [7-10](#)
 configuring, [9-15](#), [9-27](#)
 specifying, [9-24](#)
 valid values, [9-27](#)
 server pools, [6-4](#), [6-6](#), [6-7](#), [6-28](#)
 accessing, [8-1](#), [10-3](#)
 adding to a Policy Set, [9-15](#)
 adjusting size of, [6-26](#)
 and multiple databases, [7-2](#)
 and recommendations, [6-21](#)
 and services, [6-27](#)
 attributes, [6-15](#)
 candidate server lists, [6-10](#), [8-2](#)
 configuring, [7-2](#)
 constraints, [7-8](#)
 creating, [8-2](#), [9-15](#)
 directive overrides, [6-15](#)
 execute permission, [10-3](#)
 Free pool, [6-10](#), [8-2](#)
 increasing minimum pool size, [9-27](#)
 list of, [9-4](#)
 managed, [7-2](#)
 marked as managed, [8-7](#)
 maximum size of one, [7-2](#), [7-3](#), [10-3](#)
 minimum size, [6-20](#), [6-21](#), [7-10](#), [8-1](#)
 minimum size of one, [7-2](#)
 modifying settings, [9-15](#)
 modifying using Oracle Enterprise Manager, [9-15](#)
 privileges, [8-2](#)
 removing servers from, [7-2](#)
 selecting for management, [9-12](#)
 SERVER_NAMES attribute, [8-2](#)
 servers not added, [10-4](#)
 unmanageable, [10-3](#)
 unmanaged, [7-2](#)
 using multiple server pools, [6-7](#)
 viewing current settings, [9-12](#)
 SERVER_NAMES attribute, [8-2](#)
 servers

servers (*continued*)
 added to wrong server pool, [10-4](#)
 hosting the Oracle Database QoS Management server, [7-11](#)
 service levels, [6-14](#)
 SERVICE_NAME, [6-12](#)
 SERVICE_TIME, [6-5](#)
 service-level agreements (SLAs), [6-25](#), [7-10](#), [7-11](#), [9-22](#)
 services, [6-3](#), [6-27](#)
 adding to the Policy Set, [9-16](#), [9-17](#)
 and Oracle Clusterware, [7-3](#)
 and work requests, [6-27](#)
 configuring load balancing goals, [7-3](#)
 creating, [8-4](#)
 default database service, [7-3](#)
 defaults
 Performance Classes, [6-12](#)
 discovered automatically, [7-10](#)
 for database connections, [7-4](#)
 load balancing, [7-3](#)
 names, [7-5](#)
 singleton, [6-27](#)
 SINGLETON, [7-2](#), [7-3](#), [10-3](#)
 specifying multiple, [9-18](#)
 transactional shut down, [6-25](#)
 UNIFORM, [7-2](#), [7-3](#)
 used by workloads, [7-3](#)
 session attributes, [7-5](#)
 ACTION, [7-5](#)
 and classifiers, [7-5](#)
 MODULE, [7-5](#)
 program, [7-5](#)
 user name, [7-5](#)
 Set Policy, [9-26](#)
 silent mode operation
 exclude root access, [2-14](#)
 include root access, [2-14](#)
 singleton databases, [7-2](#)
 SINGLETON services, [7-3](#), [10-3](#)
 Situation Analysis, [9-9](#)
 skipped checks, [2-57](#)
 SLA
 See service-level agreements
 slice, [6-4](#)
 sockets, [22-3](#)
 SQL performance issues, [7-7](#)
 SRVCTL, [7-3](#), [9-15](#), [9-33](#), [10-3](#), [10-5](#)
 commands
 srvctl config cha, [3-10](#)
 srvctl status cha, [3-10](#)
 SSL certificates, [22-3](#)
 SSL cipher suite, [19-11](#)
 SSL protocols, [19-8](#)
 storage interconnect, [7-7](#)

subsequent email, [2-5](#)
 sudo, [2-6](#)
 support for Administrator-Managed databases,
 [6-10](#)
 supported
 connection types, [7-4](#)
 Oracle RAC database releases, [7-2](#)
 response times, [7-4](#)
 workloads, [6-26](#), [6-32](#), [7-4](#)
 supported environments, [14-1](#)
 SYSTEM
 node view, [12](#)
 system and cluster summary, [16-2](#)
 system parameters, [6-29](#)
 system performance
 evaluating, [6-18](#), [6-22](#)
 system requirements, [7-1](#), [7-6](#)

T

tabular format file output, [12](#)
 tagging, [6-12](#)
 tags, [6-11](#), [7-5](#)
 and classifiers, [7-5](#)
 for work requests, [6-12](#)

TFACTL

commands
 tfactl access, [53](#)
 tfactl analyze, [58](#)
 tfactl changes, [56](#)
 tfactl collection, [83](#)
 tfactl dbglevel, [16-11](#)
 tfactl diagcollect, [64](#)
 tfactl diagnosetfa, [50](#)
 tfactl directory, [68](#)
 tfactl events, [57](#)
 tfactl host, [50](#)
 tfactl ips, [70](#)
 tfactl ips ADD, [73](#)
 tfactl ips ADD FILE, [73](#)
 tfactl ips ADD NEW INCIDENTS
 PACKAGE, [75](#)
 tfactl ips COPY IN FILE, [74](#)
 tfactl ips CREATE PACKAGE, [76](#)
 tfactl ips DELETE PACKAGE, [78](#)
 tfactl ips FINALIZE PACKAGE, [77](#)
 tfactl ips GENERATE PACKAGE, [77](#)
 tfactl ips GET MANIFEST FROM FILE,
 [78](#)
 tfactl ips GET METADATA, [79](#)
 tfactl ips GET REMOTE KEYS FILE, [75](#)
 tfactl ips PACK, [79](#)
 tfactl ips REMOVE, [74](#)
 tfactl ips REMOVE FILE, [75](#)
 tfactl ips SET CONFIGURATION, [80](#)

TFACTL (continued)

commands (continued)
 tfactl ips SHOW CONFIGURATION, [81](#)
 tfactl ips SHOW FILES PACKAGE, [81](#)
 tfactl ips SHOW INCIDENTS PACKAGE,
 [82](#)
 tfactl ips SHOW PACKAGE, [81](#)
 tfactl ips SHOW PROBLEMS, [82](#)
 tfactl ips UNPACK FILE, [82](#)
 tfactl ips UNPACK PACKAGE, [82](#)
 tfactl ips USE REMOTE KEYS FILE, [76](#)
 tfactl managelogs, [86](#)
 tfactl print, [83](#)
 tfactl purge, [85](#)
 tfactl run, [61](#)
 tfactl set, [51](#)
 tfactl summary, [55](#)
 tfactl toolstatus, [62](#)

Oracle Trace File Analyzer command-line
 utility, [48](#)

Time Machine software, [22-13](#)

timeouts, [2-57](#)

TLS protocols, [19-8](#)

tools bundle, [16-4](#)

TOP CONSUMERS

node view, [12](#)

Trace File Analyzer

disk usage snapshots, [21-2](#)

trace files, [10-5](#)

trace logs, [12-4](#)

tracing

enabling, [10-5](#)

Troubleshoot

EXAchk, [2-53](#)

ORAchk, [2-53](#)

troubleshooting

enabling Oracle Database QoS Management
 for a cluster, [10-1](#)

enabling Oracle Database QoS Management
 for a database, [10-2](#)

U

UNIFORM services, [7-3](#)

Uninstall, [14-7](#)

unlockcells, [2-47](#)

update

Oracle Trace File Analyzer, [17-1](#)

tools bundle, [17-1](#)

upgrade phases, [22-4](#)

upload collections, [16-10](#)

user

add, [14-6](#)

remove, [14-6](#)

reset, [14-6](#)

user name, [7-5](#)
 USERENV namespace, [7-5](#)
 USERNAME, [6-12](#)
 users
 APPQOSSYS, [10-2](#)
 creating administrators, [9-33](#)
 default for QoSAdmin, [9-34](#)

V

viewing
 current server pool settings, [9-12](#)
 latest analysis results, [9-8](#)
 performance metrics, [9-29](#), [9-30](#)
 recommendation details, [9-9](#)
 recommendations, [9-8](#)
 resource wait times, [9-32](#)
 violations, [8-5](#)
 and CPU_COUNT, [8-4](#)
 configuration, [7-1](#), [7-2](#), [9-12](#)
 for Performance Objectives, [7-11](#)
 logs, [7-11](#)
 resource plan not enabled, [10-2](#)
 viewing, [9-4](#)
 VMPScan, [2-46](#)

W

wait times, [6-30](#), [7-7](#)
 for resources, [9-32](#)
 viewing, [9-4](#)
 wizards

wizards (*continued*)
 Create Policy Set, [8-7](#)
 Edit Policy Set, [9-11](#), [9-12](#), [9-18–9-20](#)
 work requests, [6-1](#), [6-11](#), [6-12](#)
 and services, [6-27](#)
 and tags, [6-12](#)
 creating classifiers, [9-18](#)
 definition, [6-26](#)
 managing separately, [6-13](#)
 metrics, [6-30](#)
 workload rank
 and database startup order, [6-18](#)
 workloads, [6-6](#), [6-19](#)
 and creating new Performance Classes, [7-4](#)
 and GV\$ view requests, [7-4](#)
 and parallel queries, [7-4](#)
 and server pools, [6-7](#)
 calls and Performance Classes, [7-5](#)
 consolidating, [6-7](#)
 database connections, [6-12](#)
 different resource requirements for work
 subsets, [7-4](#)
 managing, [7-5](#)
 monitoring, [6-5](#)
 OLTP, [7-4](#)
 scaling, [6-7](#)
 separating, [6-3](#)
 supported response times, [7-4](#)
 supported types, [6-26](#), [6-32](#), [7-4](#)
 surges, [7-9–7-11](#)
 tuning goals, [6-7](#)
 See also Performance Classes