Oracle® Autonomous Health Framework

User's Guide

12*c* Release 2 (12.2)

E63513-05

May 2017

ORACLE

Oracle Autonomous Health Framework User's Guide, 12*c* Release 2 (12.2)

E63513-05

Primary Author: Nirmal Kumar

Contributing Authors: Richard Strohm, Mark Bauer, Douglas Williams, Aparna Kamath, Janet Stern, 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

Contents

F Oracle Trace File Analyzer Command-Line and Shell Options

List of Figures

List of Tables

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

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 New Features Guide

Oracle Database Readme

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.
italic	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 Oracle Autonomous Health Framework User's Guide Release 12*c*

This preface lists changes in Oracle Autonomous Health Framework for Oracle Database 12*c* release 2 (12.2) and release 1 (12.1).

New Features for Oracle Database 12*c* Release 2 (12.2)

These are new features for Oracle Database 12*c* release 2 (12.2).

Oracle Cluster Health Advisor

Oracle Cluster Health Advisor is introduced for Oracle Database 12c release 2 (12.2). Oracle Cluster Health Advisor collects data from Oracle Real Application Clusters (Oracle RAC) and Oracle RAC One Node databases, and from operating system and hardware resources. Oracle Cluster Health Advisor then advises how to fix database or performance issues.

Enhancements to Grid Infrastructure Management Repository (GIMR)

Oracle Grid Infrastructure deployment now supports a global off-cluster Grid Infrastructure Management Repository (GIMR).

Related Topics:

Oracle Grid Infrastructure Installation and Upgrade Guide for Linux

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

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

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 (page 2-35)

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 Logon (page 2-34)

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 (page 2-38)

Viewing and Reattempting Failed Uploads

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

Related Topics:

Viewing and Reattempting Failed Uploads (page 2-40)

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 (page 2-37)

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 (page A-10)

Tracking File Attribute Changes and Comparing Snapshots (page 2-29)

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 (page 2-45)

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) (page 2-47)

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 (page 2-48)

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, "ORAchk - Health Checks for the Oracle Stack", and click the tab **Health Check Catalog** to download and view the list of health checks.

Related Topics:

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

New Features for Cluster Health Monitor 12.2.0.1.1

Two new parameters are added to the `oclumon dumpnodeview` command:

- `oclumon dumpnodeview –format csv` : This option provides CSV format output mode for dumpnodeview.
- `oclumon dumpnodeview -procag` : This option provides output of node view processes aggregated by category.

Related Topics:

OCLUMON Command Reference (page B-1)

New Features for Oracle Trace File Analyzer 12.2.0.1.1

Oracle Trace File Analyzer includes the following new features in release 12.2.0.1.1:

- Oracle Trace File Analyzer runs an automatic purge every 60 minutes to delete logs that are older than 30 days.

- Manage Automatic Diagnostic Repository (ADR) log and trace files by using the `managelogs` command.
- Oracle Trace File Analyzer now automatically monitors disk usage and records snapshots.
- Oracle Trace File Analyzer now provides event-driven Support Service Request Data Collection (SRDC) collections.
- Oracle Trace File Analyzer integrates Incident Packaging Service (IPS), and can now run IPS to show incidents, problems, and packages. IPS packages can also be included in diagnostic collection with the option to manipulate them before packaging.
- Oracle Trace File Analyzer Built on Java Runtime Environment (JRE) 1.8.

  Oracle Trace File Analyzer was built on Java Runtime Environment (JRE) 1.8 for this release. It uses the latest Java features. Bash shell is no longer required for Oracle Trace File Analyzer. Because Oracle Trace File Analyzer runs on Java instead of as a shell script, it is now supported on Microsoft Windows platforms.

  If you plan to use Oracle Trace File Analyzer, then JRE is now a requirement. JRE ships with Oracle Database and Oracle Grid Infrastructure. JRE is also included in the Oracle Trace File Analyzer Database Support Tools Bundle from My Oracle Support document 1513912.2.

  https://support.oracle.com/rs?type=doc&id=1513912.2

Related Topics:

Running On-Demand Event-Driven SRDC Collections (page 4-24)

tfactl diagcollect (page F-17)

Supported Platforms and Product Versions (page 4-6)

Oracle Grid Infrastructure Trace File Analyzer Installation (page 4-7)

Purging Oracle Trace File Analyzer Logs Automatically (page 4-34)

Managing Automatic Diagnostic Repository Log and Trace Files (page 4-33)

Managing Disk Usage Snapshots (page 4-34)

New Features for Hang Manager

Hang Manager includes the following new features:

- Sensitivity Setting

  Adjust the threshold period that Hang Manager waits to confirm if a session is hung by setting the `sensitivity` parameter.
- Number of Trace Files Setting

  Adjust the number of trace files that can be generated within the trace file sets by setting the `base_file_set_count` parameter.
- Size of Trace File Setting

  Adjust the size (in bytes) of trace files by setting the `base_file_size_limit` parameter.

Related Topics:

Optional Configuration for Hang Manager (page 7-3)

New Features for Memory Guard

- Alert Notifications When Memory Pressure is Detected

  Memory Guard now sends alert notifications if Memory Guard finds any server at risk. Find those notifications in the audit logs.

Related Topics:

Use of Memory Guard in Oracle Real Application Clusters (Oracle RAC) Deployment (page 6-3)

New Features for Oracle Database Quality of Service Management 12c Release 2 (12.2.0.1)

- New `qosmserver` to Replace OC4J J2EE Container

  In earlier releases, Oracle Database Quality of Service Management Server was deployed in an OC4J J2EE container. OC4J J2EE is not supported on the latest versions of Java, and had a greater resource footprint than needed by Oracle Database Quality of Service Management. A profiled version of Tomcat, known as the `qosmserver`, replaces the OC4J J2EE container.

- Full Support for Administrator-Managed and Multitenant Oracle RAC Databases

  In Oracle Database 12*c* release 1 (12.1), Oracle Database Quality of Service Management supported administrator-managed Oracle RAC and Oracle RAC One Node databases in its Measure-Only and Monitor modes. In this release, you can use Oracle Database Quality of Service Management support in Management mode for administrator-managed Oracle RAC databases and Multitenant Oracle RAC Databases. However, Oracle Database Quality of Service Management cannot expand or shrink the number of instances by changing the server pool size for administrator-managed databases because administrator-managed databases do not run in server pools. Oracle Enterprise Manager Cloud Control supports this new feature in the Oracle Database Quality of Service Management pages.

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 12*c* release 1 (12.1). In Oracle Database 12*c* 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.

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.

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 (page 1-5)
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 (page 1-5)
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 Collector (page 1-6)
Oracle Trace File Analyzer Collector 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 (page 1-6)
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 (page 1-7)
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 (page 1-8)
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 (page 1-8)
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:

Analyzing Risks and Complying with Best Practices (page 2-1)

Oracle ORAchk and Oracle EXAchk Command-Line Options (page A-1)

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 (page 3-1)

1.2.3 Introduction to Oracle Trace File Analyzer Collector

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

Related Topics:

Collecting Diagnostic Data and Triaging, Diagnosing, and Resolving Issues (page 4-1)

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:

Proactively Detecting and Diagnosing Performance Issues for Oracle RAC (page 5-1)

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 (page 6-1)

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 (page 7-1)

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 (page 9-1)

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.

This chapter describes how to use Oracle ORAchk or Oracle EXAchk and contains the following sections:

Integrating Health Check Results with Other Tools (page 2-48)
Integrate health check results from Oracle ORAchk and Oracle EXAchk into Enterprise Manager and other third-party tools.

Troubleshooting Oracle ORAchk and Oracle EXAchk (page 2-53)
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 (page 1-5)

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.

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 (page 2-17)

Starting and Stopping the Daemon (page 2-16)

Querying the Status and Next Planned Daemon Run (page 2-26)

Configuring the Daemon for Automatic Restart (page 2-17)

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 (page 2-3)
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? (page 2-3)
Health check reports contain the health status of each system grouped under different sections of the report.

Subsequent Email Notifications (page 2-5)
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 (page 2-15)

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 (page 2-6)
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 (page 2-7)
Handling of `root` passwords depends on whether you have installed the Expect utility.

Configuring Email Notification System (page 2-8)
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 temporary directory, that is, `$HOME` used by Oracle ORAchk and Oracle EXAchk. Change the temporary directory by setting the environment variable `RAT_TMPDIR`.

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 temporary directory, `$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/.orachk/root_orachk.sh

     user ALL=(root) NOPASSWD:/root/.exachk/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 the `root` password is not the same for all the components, then the Expect utility prompts you to validate the `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.

See Also:

http://expect.sourceforge.net/

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

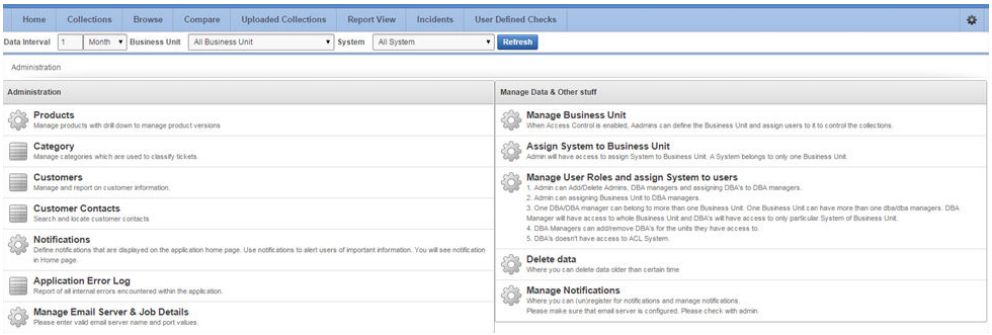
Note:

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

To configure the email notification system:

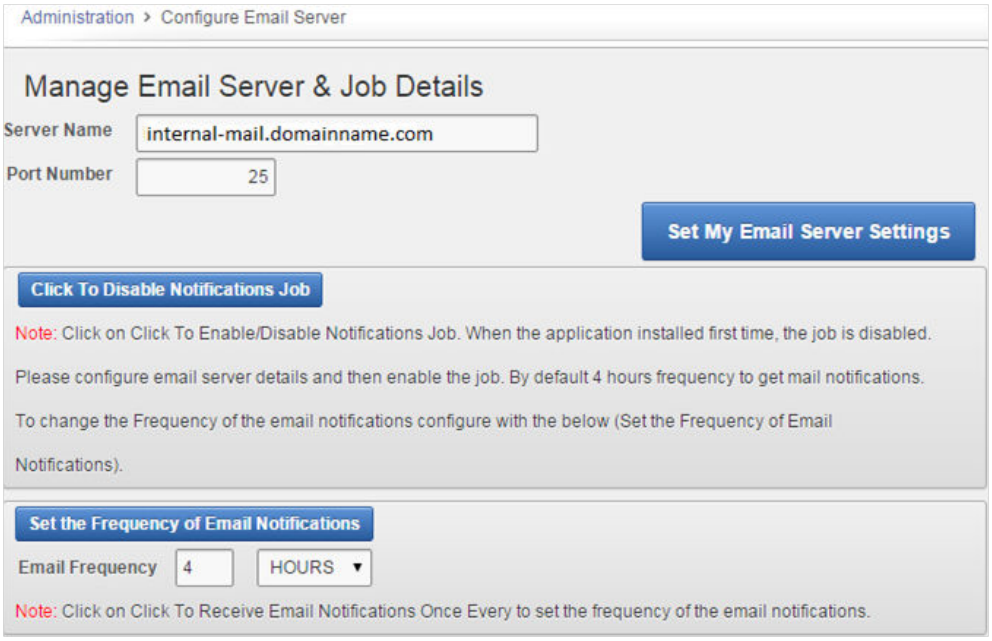
1. Log in to Oracle Health Check Collections Manager, and then click **Administration** at the upper-right corner.

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

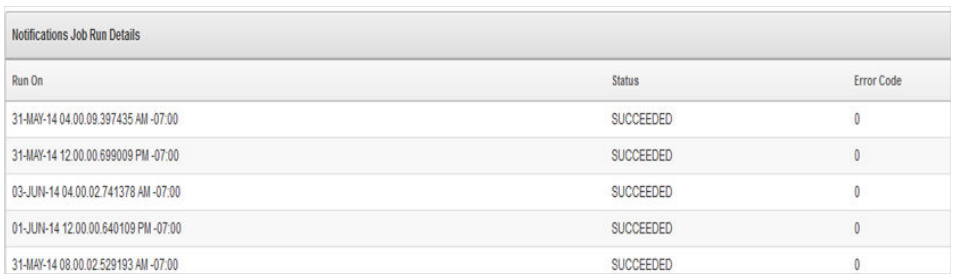


a. Specify a valid **Email Server Name**, **Port Number**, and then click **Set My Email Server Settings**.

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

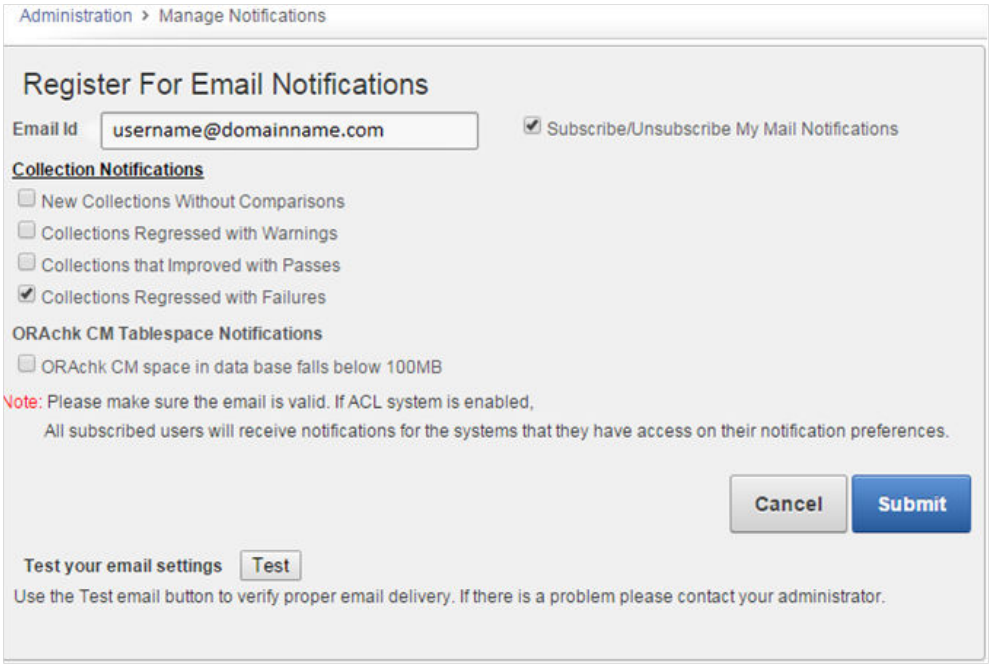


Notifications Job Run Details

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

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

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



a. If you are configuring for the first time, then enter your email address.

   Subsequent access to **Manage Notifications** page shows your email address automatically.

b. By default, **Subscribe/Unsubscribe My Mail Notifications** is checked. Leave as is.

c. Under **Collection Notifications**, choose the type of collections for which you want to receive notifications.

d. Select to receive notification when the available space in ORAchk CM Tablespace falls below 100 MB.

e. 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**.

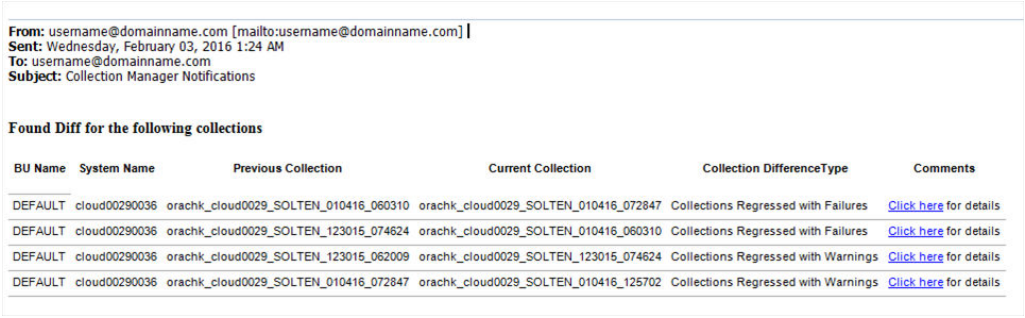
Note:

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



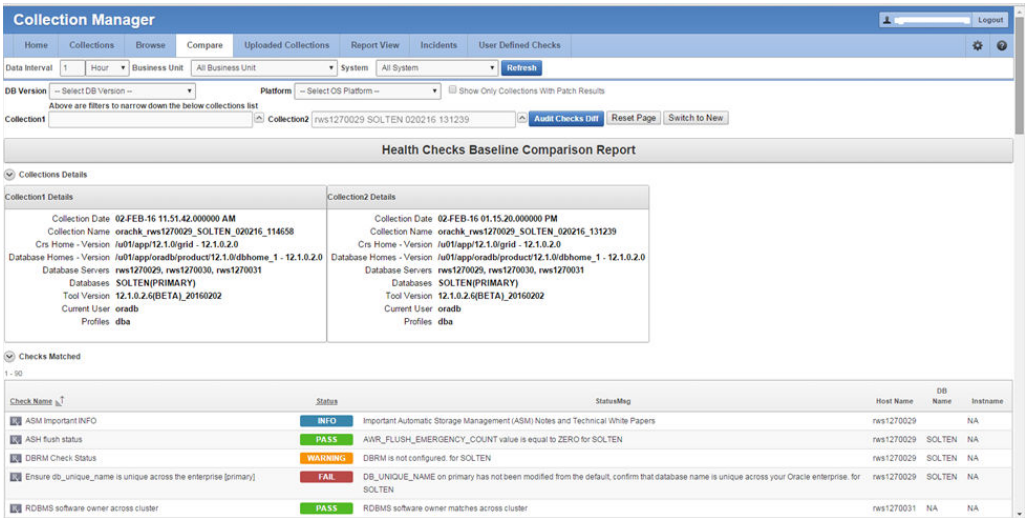
From: username@domainname.com [mailto:username@domainname.com]
Sent: Wednesday, February 03, 2016 1:24 AM
To: username@domainname.com
Subject: Collection Manager Notifications

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



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 (page 2-12)
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 (page 2-14)
Run health checks automatically by scheduling them with the Automated Daemon Mode operation.

Running On-Demand With or Without the Daemon (page 2-15)
When running on-demand, if the daemon is running all prompts, then the daemon answers where possible including the passwords.

Generating a Diff Report (page 2-15)
The diff report attached to the previous email notification shows a summary of differences between the most recent runs.

Sending Results by Email (page 2-15)
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.

See Also:

http://expect.sourceforge.net/

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 (page 2-15)

Sending Results by Email (page 2-15)

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 (page 2-2)

Running Health Checks On-Demand (page 2-12)

2.4.3 Running On-Demand With or Without the Daemon

When running on-demand, if the daemon is running all prompts, then the daemon answers 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, use the `-daemon` option.

   ```
   $ ./orachk -daemon

   $ ./exachk -daemon
   ```

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:

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 (page 2-16)
Start and stop the daemon and force the daemon to stop a health check run.

Configuring the Daemon for Automatic Restart (page 2-17)
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 (page 2-17)
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 (page 2-26)
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.

Restarting the daemon automatically requires passwordless SSH user equivalence to `root` for the user who is configuring the auto-start feature, for example, `root` or `oracle`. If passwordless SSH user equivalence is not in place, then Oracle ORAchk and Oracle EXAchk optionally configure for you.

The passwordless SSH user equivalence is retained as long as the daemon automatic restart functionality is configured.

Deconfiguring the daemon automatic restart feature restores the SSH configuration to the state it was found before automatic restart was configured.

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.

   Pre-configure `root` user equivalence for all `COMPUTE`, `STORAGE`, or `IBSWITCHES` using the `-initpresetup` option (`root` equivalency for `COMPUTE` nodes is mandatory for setting up the automatic restart functionality):

   ```
   $ ./orachk -initpresetup

   $ ./exachk -initpresetup
   ```

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 (page 2-18)
   Schedule recurring health check runs using the AUTORUN_SCHEDULE daemon option.

   AUTORUN_FLAGS (page 2-19)
   The AUTORUN_FLAGS daemon option determines how health checks are run.

   NOTIFICATION_EMAIL (page 2-20)
   Set the NOTIFICATION_EMAIL daemon option to send email notifications to the recipients you specify.

   collection_retention (page 2-20)
   Set the collection_retention daemon option to purge health check collection results that are older than a specified number of days.

   PASSWORD_CHECK_INTERVAL (page 2-21)
   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.

   AUTORUN_INTERVAL (page 2-21)
   The AUTORUN_INTERVAL daemon option provides an alternative method of regularly running health checks.

   Setting Multiple Option Profiles for the Daemon (page 2-22)
   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 (page 2-23)
   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 day month day_of_week
   ```

where:

- *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 =* * * *"`	Runs every hour.
`"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 =8,20 * * 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 syadmin"`	Runs only the `dba` profile checks and tags the output with the value `sysadmin`.

Table 2-2 (Cont.) AUTORUN_FLAGS

Example	Result
`-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.

Note:

Specify the `collection_retention` option in lower case.

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 AUTORUN_INTERVAL

The AUTORUN_INTERVAL daemon option provides an alternative method of regularly running health checks.

Note:

The AUTORUN_SCHEDULE option supersedes the AUTORUN_INTERVAL option. The AUTORUN_INTERVAL option is retained for backwards compatibility. Oracle recommends that you use the AUTORUN_SCHEDULE option.

To configure recurring health check runs:

1. Set the AUTORUN_INTERVAL option, as follows:

   ```
   AUTORUN_INTERVAL=n [d | h]
   ```

   where:

   - *n* is a number
   - *d* is days
   - *h* is hours

Table 2-3 AUTORUN_INTERVAL

Example	Result
"AUTORUN_INTERVAL =1h"	Runs every hour.
"AUTORUN_INTERVAL =12h"	Runs every 12 hours.
"AUTORUN_INTERVAL =1d"	Runs every day.
"AUTORUN_INTERVAL =7d"	Runs every week.

2.5.3.7 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]
```

2.5.3.8 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_retension = 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_retension = 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.

> **See Also:** *Oracle ORAchk and EXAchk User's Guide* for more information about Oracle Health Check Collections Manager.

- 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

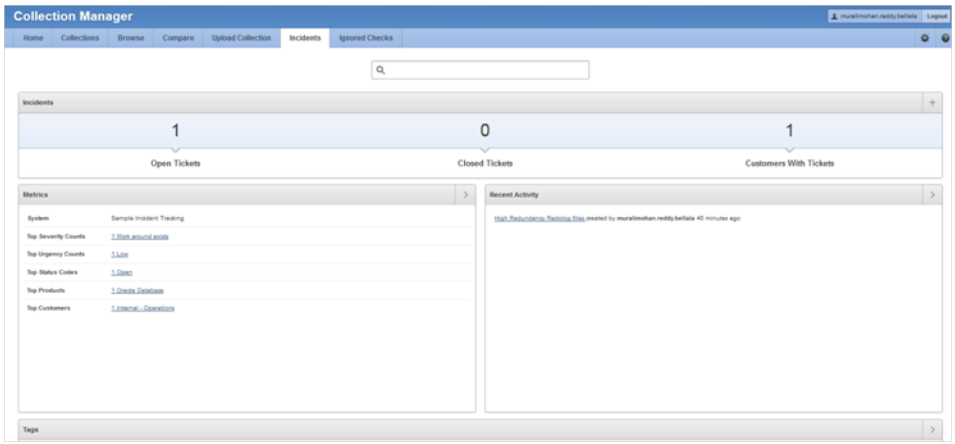
- The **Browse** tab enables you to create a new ticket on individual checks
- The **Collections** tab enables you to create a single ticket for entire the collection
- Delta (Δ) symbol is color coded red, blue, and green based on the ticket status
  - **RED (No Incident ticket exists)**: Initiates the 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 (page 2-46)

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.

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]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.

See Also: *Oracle ORAchk and EXAchk User's Guide* for more information about Oracle Health Check Collections Manager.

Selectively Capturing Users During Logon (page 2-34)
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 (page 2-35)
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 (page 2-37)
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.

Uploading Collections Automatically (page 2-38)
Configure Oracle ORAchk and Oracle EXAchk to upload check results automatically to the Oracle Health Check Collections Manager database.

Viewing and Reattempting Failed Uploads (page 2-40)
Configure Oracle ORAchk and Oracle EXAchk to display and reattempt to upload the failed uploads.

Authoring User-Defined Checks (page 2-41)
Define, test, and maintain your own checks that are specific to your environment.

Finding Which Checks Require Privileged Users (page 2-45)
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 (page 2-46)
Create or edit incident tickets for individual checks or for an entire collection.

Viewing Clusterwide Linux Operating System Health Check (VMPScan) (page 2-47)
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.

2.8.1 Selectively Capturing Users During Logon

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

Automatically capturing users during logon 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:

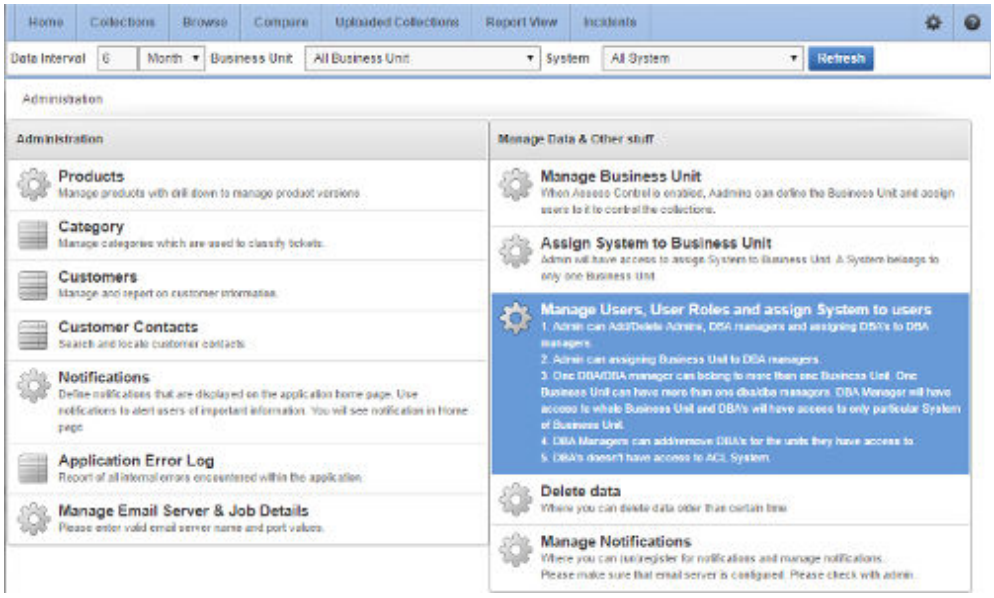
The Oracle Health Check Collections Manager roles are specific to Oracle Health Check Collections Manager and do not equate to system privileges. For example, the DBA role is not granted SYSDBA system privilege.

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:

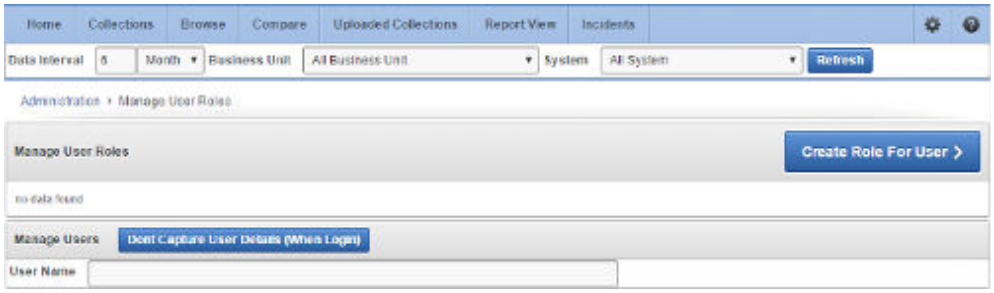
1. Click **Administration**, and then select **Manage Users, User Roles and assign System to users**.

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



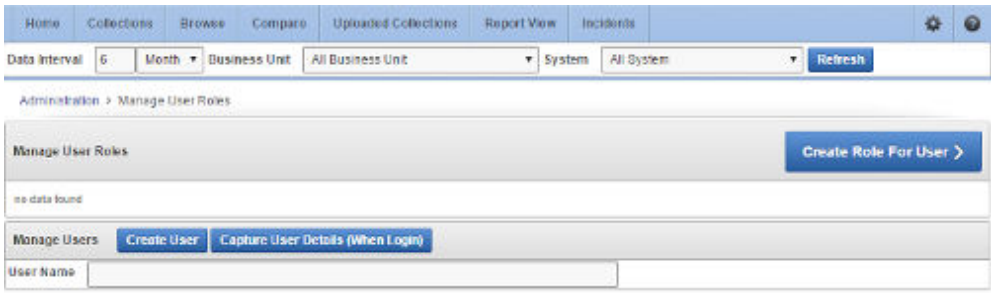
2. To disable automatic capture of users details, click **Don't Capture User Details (When Login)**.

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)



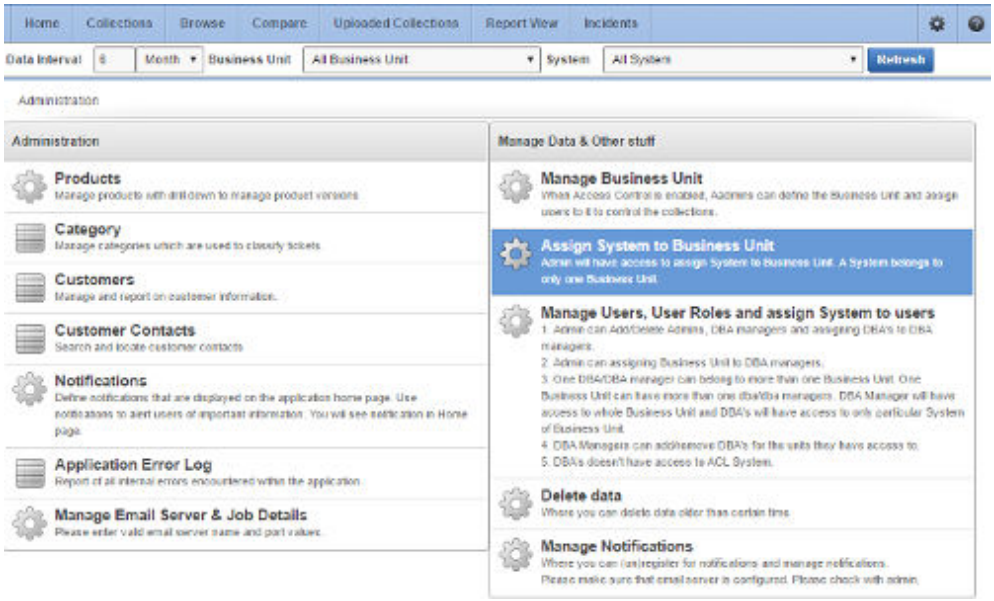
2.8.2 Bulk Mapping Systems to Business Units

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:

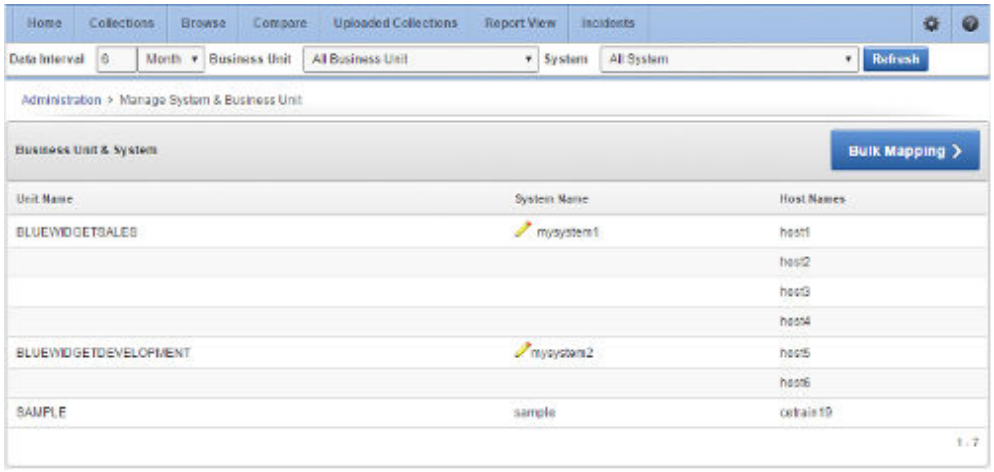
1. Click **Administration**, then select **Assign System to Business Unit**.

Figure 2-11 Assign System to Business Unit



2. Click **Bulk Mapping**.

Figure 2-12 Bulk Mapping

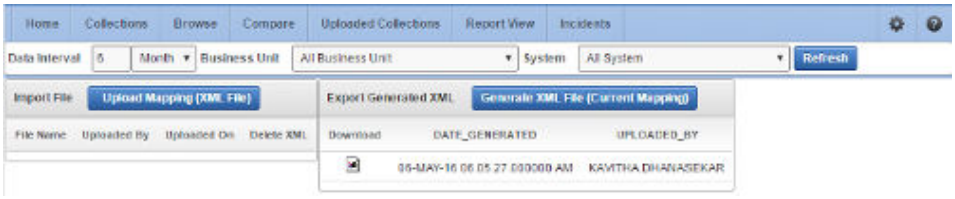


3. Upload a mapping XML.

   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



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

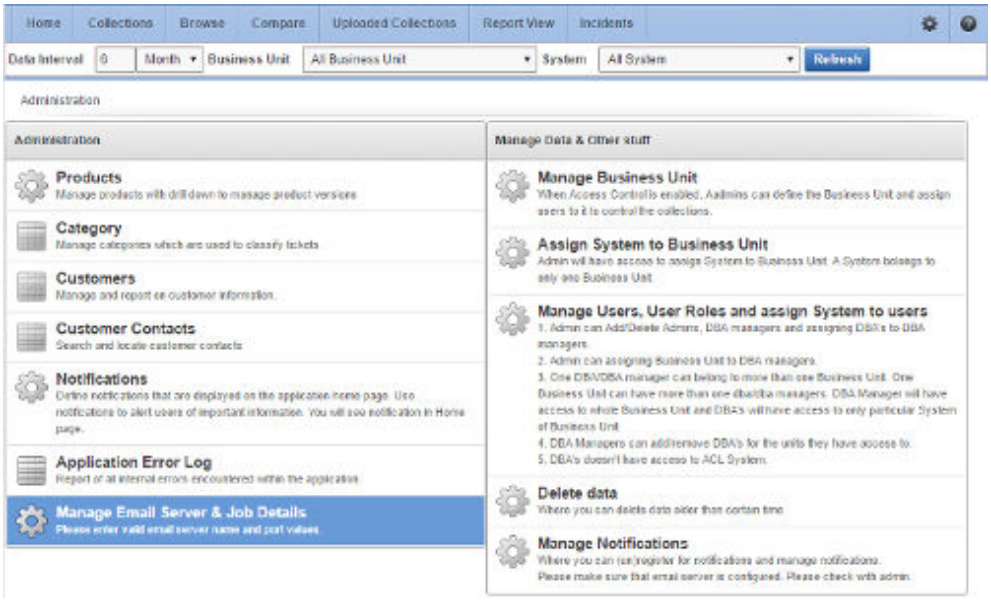
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.

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**.

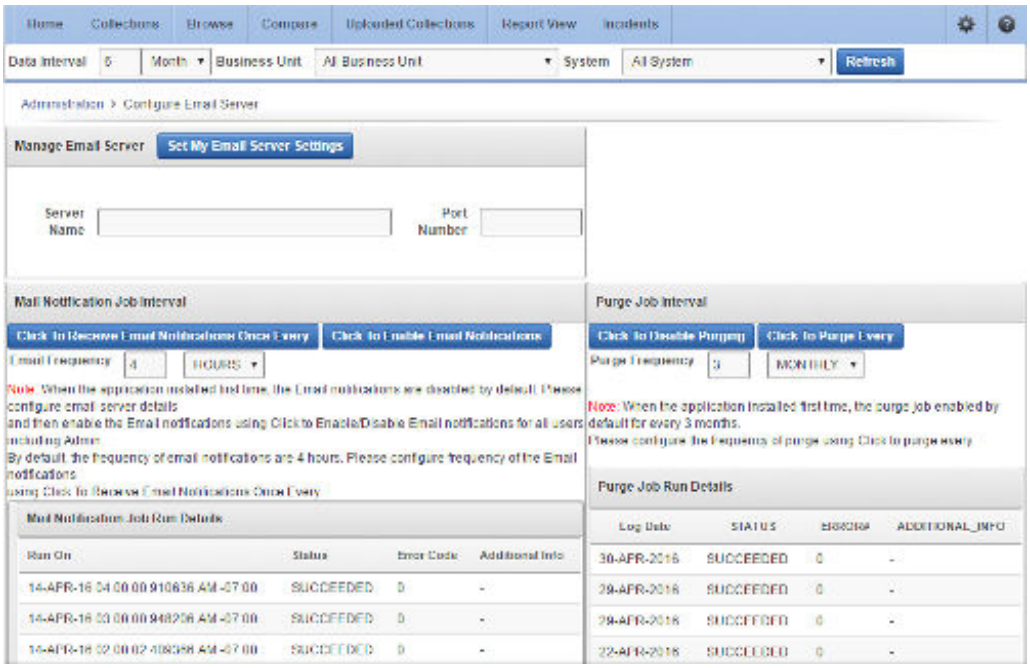
Figure 2-14 Manage Email Server and Job Details



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.

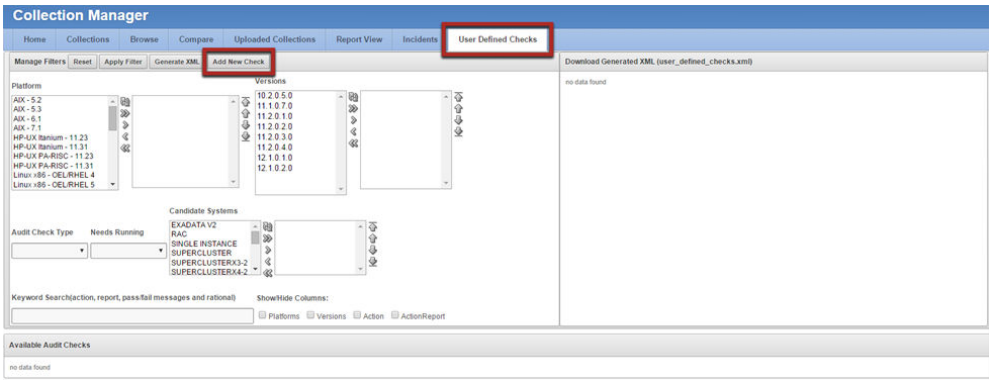
Oracle supports the framework for creating and running user-defined checks, but not the logic of the checks. It is your responsibility to test, verify, author, maintain, and support user-defined checks. At runtime, Oracle ORAchk and Oracle EXAchk script run the user-defined checks and display the results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

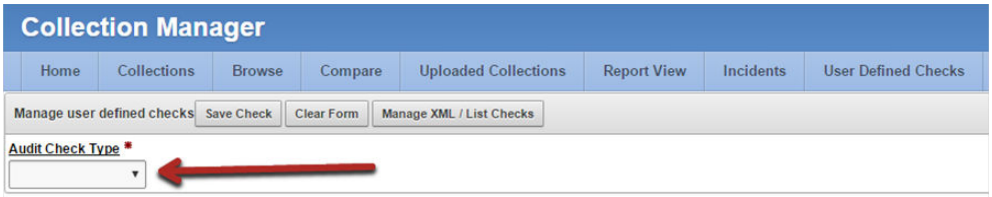
Figure 2-16 User-Defined Checks Tab



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

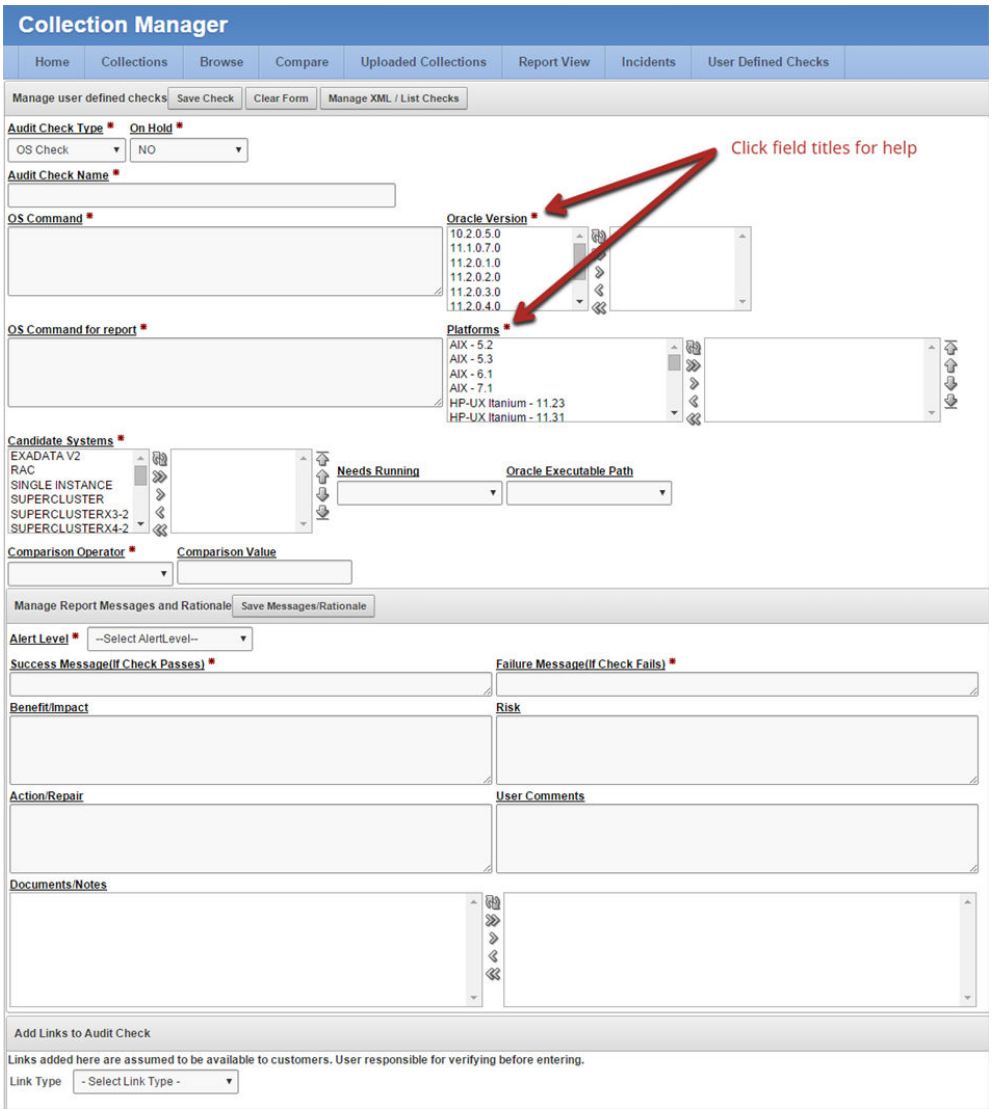


Once you have selected an **Audit Check Type**, Oracle Health Check Collections Manager updates the applicable fields.

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

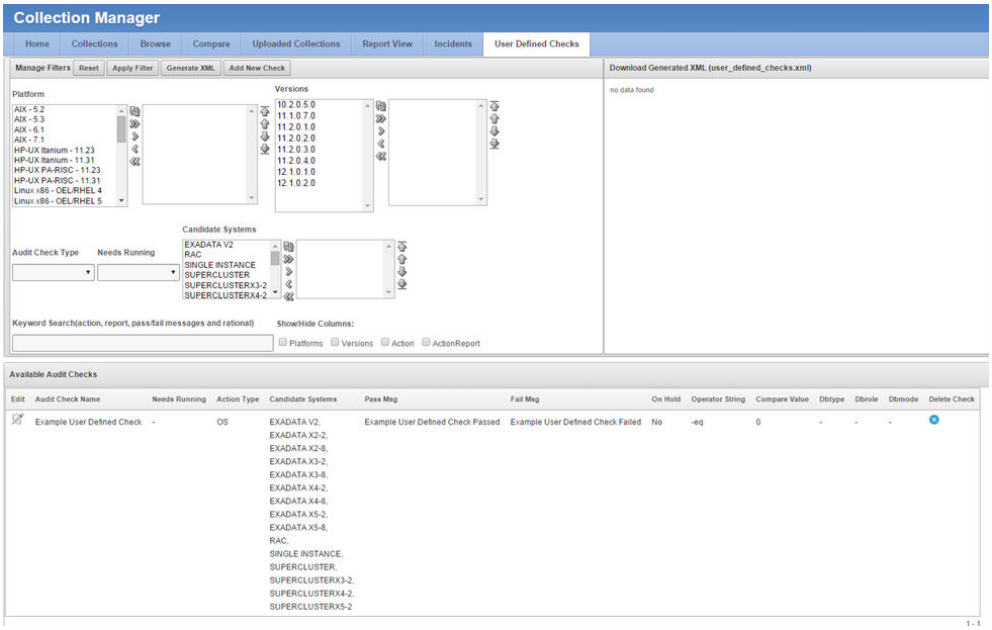
Operating system and SQL commands are supported. Running user-defined checks as `root` is **NOT** supported.

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



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**.

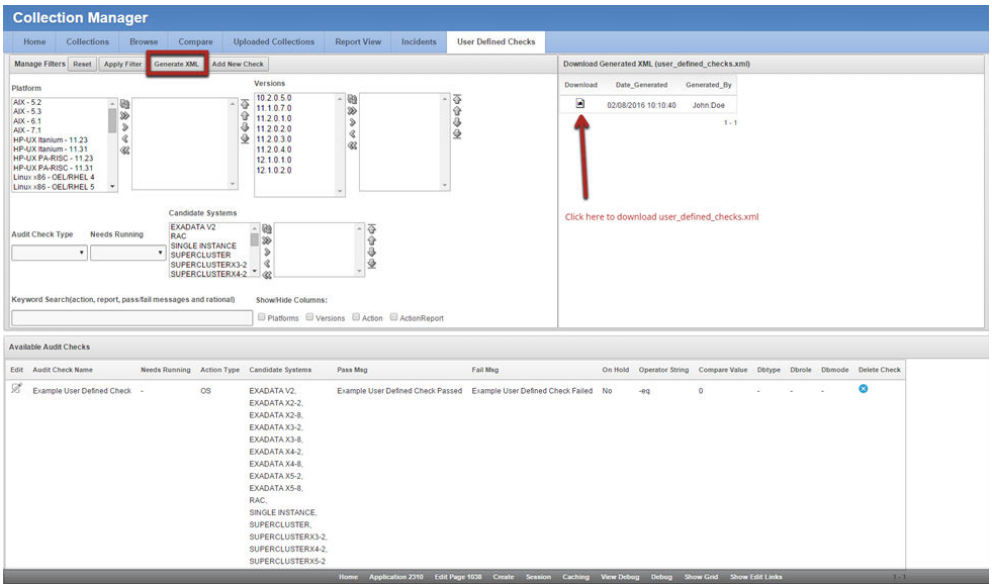
   On the right, find a link to download the generated `user_defined_checks.xml` file.

   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



5. 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
   ```

6. 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:

1. Go to My Oracle Support note 1268927.2.

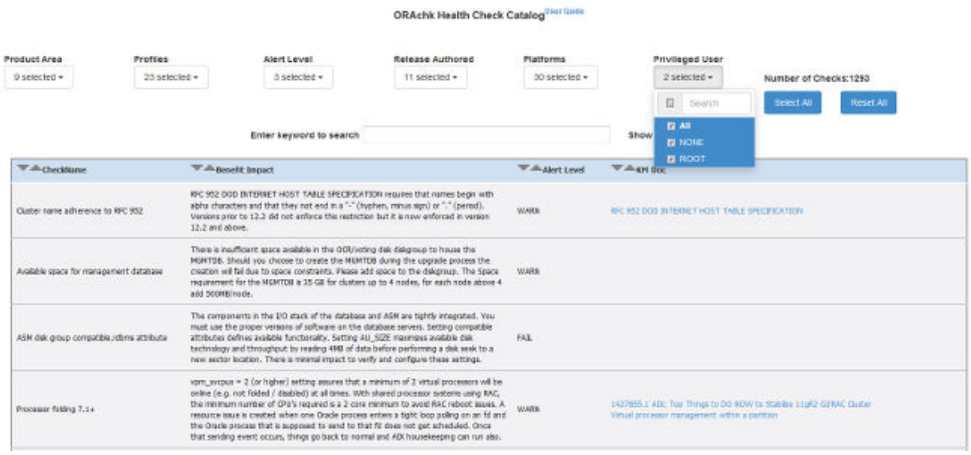
   https://support.oracle.com/rs?type=doc&id=1268927.2

2. Click the **Health Check Catalog** tab.

3. Click **Open ORAchk Health Check Catalog** to open or download the `ORAchk_Health_Check_Catalog.html` file.

4. Click the **Privileged User** drop-down list and then clear or select the checkboxes appropriately.

Figure 2-21 Oracle ORAchk - Privileged User



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 (page 2-46)

Editing Incident Tickets (page 2-46)

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**.

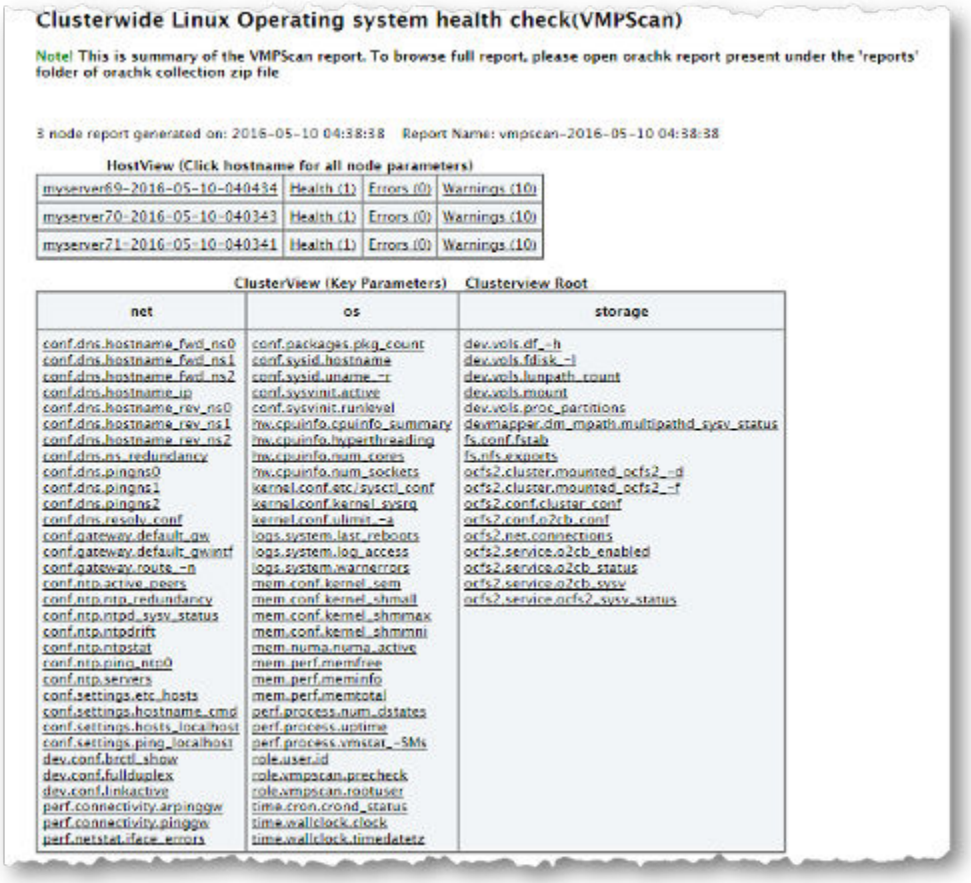
> **Note:** Click the delta symbol colored GREEN in the **Collections** or **Browse** tabs to edit incident tickets.

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 (page 2-51)
Integrate health check results from Oracle ORAchk and Oracle EXAchk into Oracle Enterprise Manager.

Integrating Health Check Results with Third-Party Tool (page 2-48)
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 (page 2-49)
Oracle ORAchk and Oracle EXAchk upload collection results from multiple instances into a single database for easier consumption of check results across your enterprise.

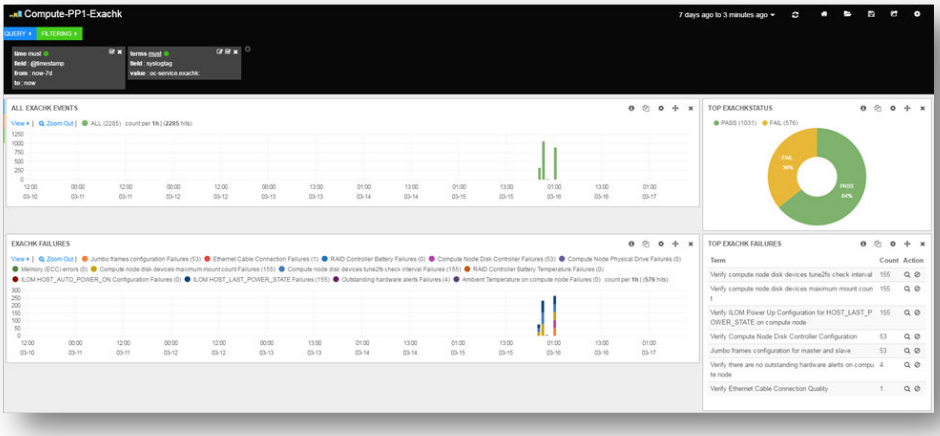
2.10.2 Integrating Health Check Results with Third-Party Tool

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

See Also:

- https://www.elastic.co/products/elasticsearch
- https://www.elastic.co/products/kibana

Figure 2-23 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.

See Also:

See "Logging Alerts to the syslogd Daemon" for more details about configuring the types of messages to log and their output location.

https://docs.oracle.com/cd/E19424-01/820-4809/log_syslog/index.html

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 2-4 Uploading Collection Results into a Database

Table	What Get's Uploaded
`rca13_docs`	Full zipped collection results.
`auditcheck_result`	Health check results.
`auditcheck_patch_result`	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 (page 2-38)

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 12*c* release 12.1.0.5 and earlier releases.

See Also:

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

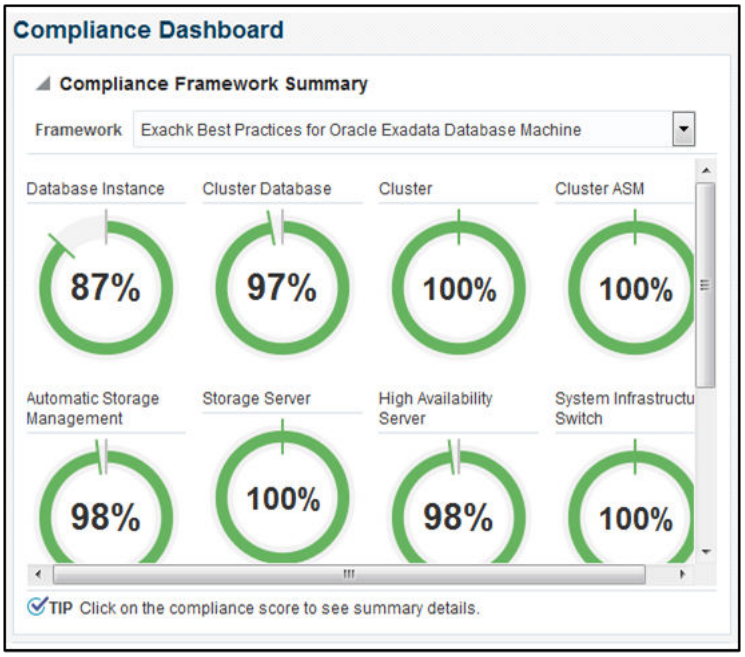
---

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 2-24 Compliance Dashboard



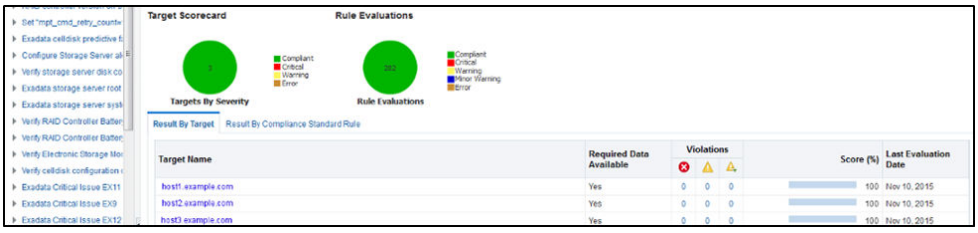
- Related checks are grouped into compliance standards where you can view targets checked, violations, and average score.

Figure 2-25 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 Systems Infrastructure Switch Best Practices For Oracle Exadata Database Machine	Systems 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	99
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 2-26 Compliance Standards Drill-Down



See Also:

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

Oracle Enterprise Manager Licensing Information User Manual for more details.

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.

See Also:

Oracle ORAchk and EXAchk User's Guide for more information about troubleshooting Oracle ORAchk and Oracle EXAchk.

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:
  https://support.oracle.com/rs?type=doc&id=1268927.2
- For Oracle EXAchk, refer to My Oracle Support Note 1070954.1:
  https://support.oracle.com/rs?type=doc&id=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.

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
   ```

   For example:

   ```
   $ ./orachk -debug
   + PS4='$(date "+ $LINENO: + ")'
    36276: + [[ -z 1 ]]
    36302: + sed 's/[\.\/]//g'
   ```

```
 36302: + basename /global/u01/app/oracle/arch03/ORACLE_CHECK/ORACLE_SR/orachk
 36302: + echo orachk
 36302: + program_name=orachk
 36303: + which bash
 36303: + echo 0
 36303: + bash_found=0
 36304: + SSH_PASS_STATUS=0
 36307: + set +u
 36309: + '[' 0 -ne 0 ']'
 36315: + raccheck_deprecate_msg='RACcheck has been deprecated.  ORAchk provides
the same functionality.
 Please switch to using ORAchk from same directory.\n\nRACcheck will not be
available after this (12.1.0.2.3) release.
 See MOS Note "RACcheck Configuration Audit Tool Statement of Direction - name
change to ORAchk (Doc ID 1591208.1)".\n'
 36316: + '[' orachk = raccheck ']'
 36325: + export LC_ALL=C
 36325: + LC_ALL=C
 36326: + NO_WRITE_PASS=0
 36327: + ECHO=:
 36328: + DEBUG=:
 36329: + AUDITTAB=db_audit
 36379: + supported_modules='PREUPGR
 . . . . . .
 . . . . . .
```

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 debug output file contains:

- `bash -x` of program on local node
- `bash -x` of program on all remote nodes
- `bash -x` of all dynamically generated and called scripts
  - 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 the daemon with `-d start_debug` option generates both debug for daemon and includes 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_SSHELL` environment variable pointing to the location of SSH:

     ```
     $ export RAT_SSHELL=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_SCOPY` environment variable pointing to the location of SCP:

     ```
     $ export RAT_SCOPY=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  zip file 2 > zip file 3 > zip file ...

./exachk –merge zipfile 1  zip file 2 > zip file 3 > zip file ...
```

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
```

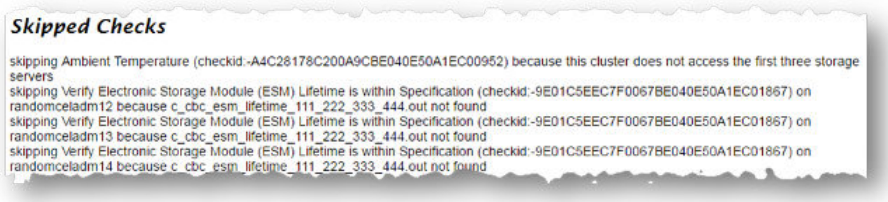
In which case, you must run as `root` or unzip again as the Oracle software install user.

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



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 2-5 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

Collecting Operating System Resources Metrics

Use Cluster Health Monitor to collect diagnostic data to analyze performance degradation or failures of critical operating system resources.

This chapter describes how to use Cluster Health Monitor and contains the following sections:

Related Topics:

Introduction to Cluster Health Monitor (page 1-5)

3.1 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

3.2 Collecting Cluster Health Monitor Data

Collect Cluster Health Monitor data from any node in the cluster by running the *Grid_home*/bin/diagcollection.pl script on the node.

When an Oracle Clusterware error occurs, run the diagcollection.pl diagnostics collection script to collect diagnostic information from Oracle Clusterware into trace files.

Run the diagcollection.pl script as root from the Grid_home/bin directory.

Note:

- Oracle recommends that you run the diagcollection.pl script on all nodes in the cluster to collect Cluster Health Monitor data. Running the script on all nodes ensures that you gather all information needed for analysis.
- Run the diagcollection.pl script as a root privileged user.

To run the data collection script only on the node where the cluster logger service is running:

1. Run the command $ *Grid_home*/bin/oclumon manage -get master.
2. Log in as a user with xx privilege, and change directory to a writable directory outside the Grid home.
3. Run the command diagcollection.pl --collect.

   For example:

   Linux:

   ```
   $ Grid_home/bin/diagcollection.pl --collect
   ```

   Microsoft Windows:

   ```
   C:\Grid_home\perl\bin\perl.exe
   C:\Grid_home\bin\diagcollection.pl --collect
   ```

   Running the command mentioned earlier collects all the available data in the Oracle Grid Infrastructure Management repository, and creates a file using the format chmosData_host_name_time_stamp.tar.gz.

   For example: chmosData_stact29_20121006_2321.tar.gz.
4. Run the command $ *Grid_home*/bin/diagcollection.pl --collect --chmos --incidenttime *time* --incidentduration *duration* to limit the amount of data collected.

In the command mentioned earlier, the format for the `--incidenttime` argument is `MM/DD/YYYY24HH:MM:SS` and the format for the `--incidentduration` argument is `HH:MM`.

For example:

```
$ Grid_home/bin/diagcollection.pl --collect --crshome Grid_home
   --chmos --incidenttime 07/21/2013 01:00:00 --incidentduration 00:30
```

Related Topics:

Diagnostics Collection Script (page C-1)

tfactl diagcollect (page F-17)

Oracle Trace File Analyzer Command-Line and Shell Options (page F-1)

Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections (page 4-5)

Diagnostics Collection Script (page C-1)

3.3 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.

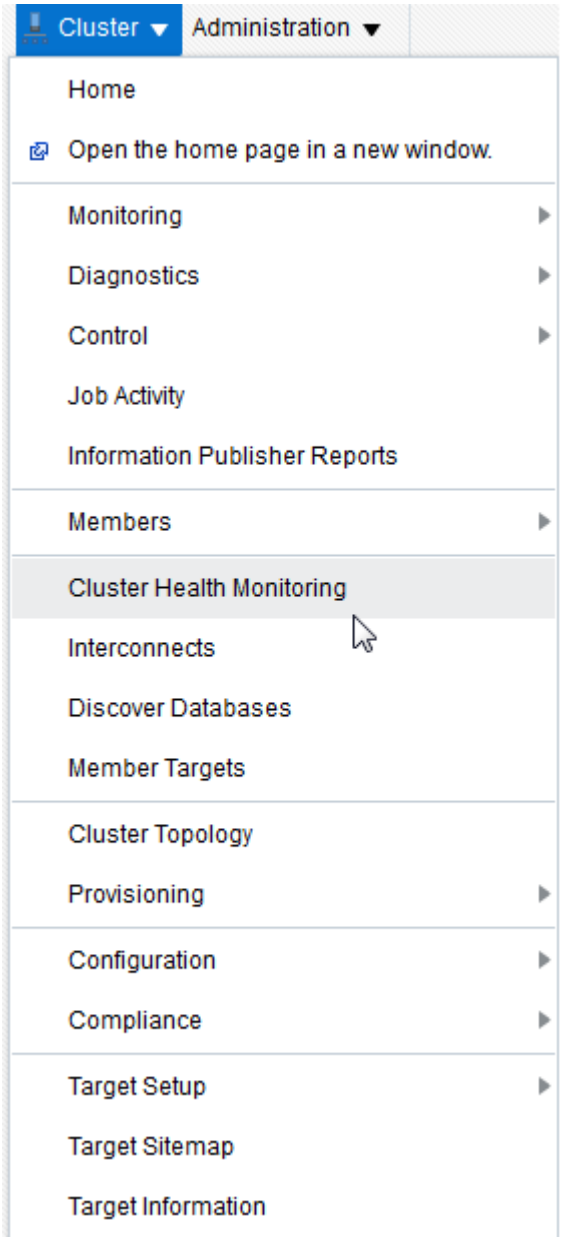
Each category is able to be separately display in greater detail showing more metrics. For example, selecting CPU results in cluster graphs detailing CPU System Usage, CPU User Usage, and CPU Cue Length. From any cluster view, you can select individual node views to more closely examine performance of 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 Gird 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 3-1 EMCC - Cluster Health Monitoring

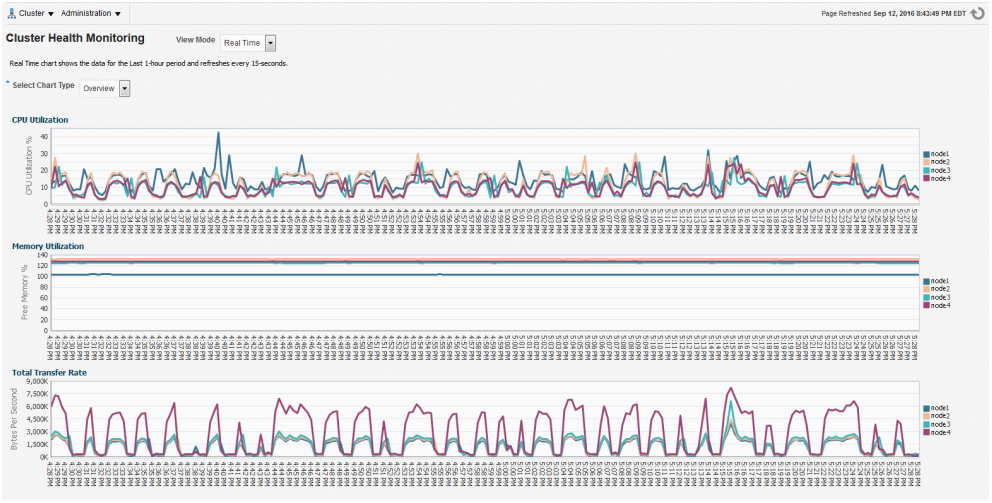


4. Enter Cluster Health Monitor login credentials.
5. From the **View Mode** drop-down list, select the **Real Time** option to view the current data.

   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 3-2 Cluster Health Monitoring - Real Time Data



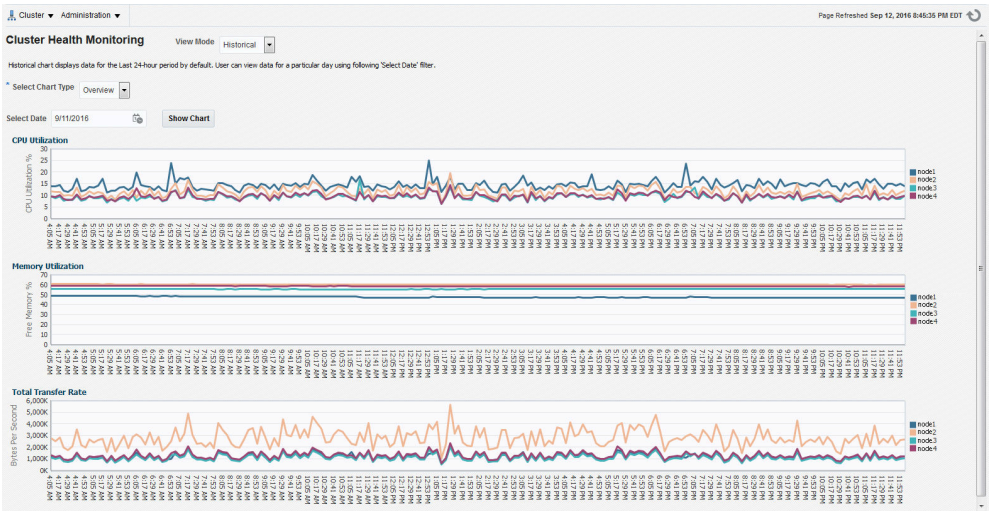
6. From the **View Mode** drop-down list, select the **Historical** option to view data for the last 24 hours.

   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 3-3 Cluster Health Monitoring - Historical Data



4

Collecting Diagnostic Data and Triaging, Diagnosing, and Resolving Issues

Use Oracle Trace File Analyzer to collect comprehensive diagnostic data that saves your time and money.

This chapter describes how to use Oracle Trace File Analyzer Collector and contains the following sections:

Related Topics:

Introduction to Oracle Trace File Analyzer Collector (page 1-6)

4.1 Understanding Oracle Trace File Analyzer

Oracle Trace File Analyzer Collector and Oracle Trace File Analyzer simplify collecting diagnostic data and resolving issues.

Oracle Trace File Analyzer Collector does the following:

- Automatically detects significant Oracle database and Oracle Grid Infrastructure problems
- Executes diagnostics and collection of log files
- Trims log files around relevant time periods
- Coordinates collection around the cluster
- Packages all diagnostics in a single package on a single node
- Provides `tfactl` command-line interface and shell that simplifies the usage of the database support tools

Oracle Trace File Analyzer uses data collected by Oracle Trace File Analyzer Collector to provide the following:

- Summary report of configured systems, changes, events, and system health
- Analysis of common error log messages

Oracle Trace File Analyzer Architecture (page 4-2)

Oracle Trace File Analyzer Collector Automated Diagnostic Collections (page 4-3)

When running in daemon mode, Oracle Trace File Analyzer Collector monitors important Oracle logs for events symptomatic of a significant problem.

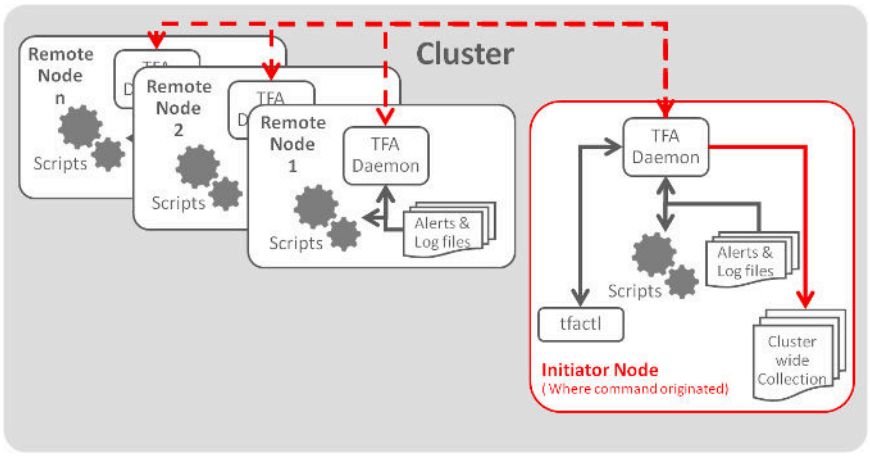
Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections (page 4-5)

Use Oracle Trace File Analyzer Collector to request an on-demand collection if you have determined a problem has occurred, and you are not able to resolve.

4.1.1 Oracle Trace File Analyzer Architecture

Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector use a single daemon on the database server. If the database is clustered, then a daemon runs on each node of the cluster.

Figure 4-1 Oracle Trace File Analyzer Architecture



Control Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector through the command-line interface `tfactl`, which can either be used in single command fashion or as a command shell.

The `tfactl` command communicates with the local daemon, which then coordinates with all daemons in the cluster. Each daemon runs the necessary diagnostic scripts locally, collects, and then trims local logs.

All daemons coordinate to create the resulting cluster-wide collection on the node where the `tfactl` command was run. If the collection was initiated automatically, then the email notification contains the location of the cluster-wide collection.

4.1.2 Oracle Trace File Analyzer Collector Automated Diagnostic Collections

When running in daemon mode, Oracle Trace File Analyzer Collector monitors important Oracle logs for events symptomatic of a significant problem.

Based on the event type detected, Oracle Trace File Analyzer then starts an automatic diagnostic collection.

The data collected depends on the event detected. Oracle Trace File Analyzer coordinates the collection around the cluster, and trims the logs around relevant time periods, and then packs all collection results into a single package on one node.

Oracle Trace File Analyzer does not do a collection for every event detected. When an event is first identified, Oracle Trace File Analyzer triggers the start point for a collection and then waits for five minutes before starting diagnostic gathering. The purpose of waiting for five minutes is to capture any other relevant events together.

- If events are still occurring after 5 minutes, then Oracle Trace File Analyzer waits to complete diagnostic collection for up to a further five minutes for 30 seconds with no events occurring.
- If events are still occurring 10 minutes after first detection, then Oracle Trace File Analyzer forces a diagnostic collection and generates a new collection start point for the next event.

Once the collection is complete, Oracle Trace File Analyzer sends email notification that includes the details of where the collection results are, to the relevant recipients.

Figure 4-2 Automatic Diagnostic Collections

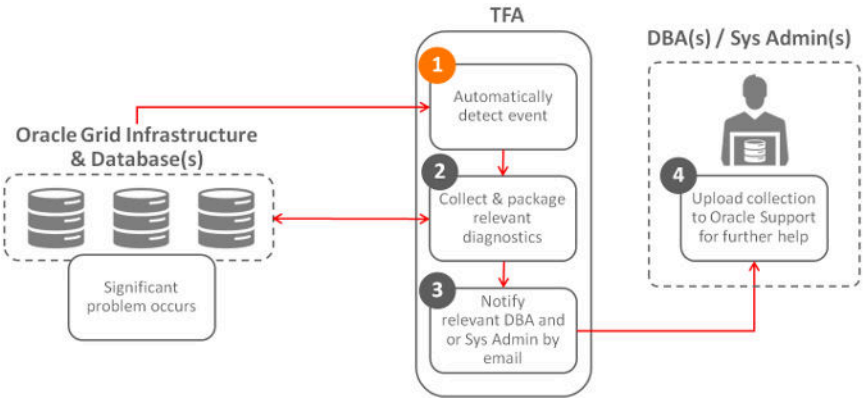


Table 4-1 Trigger Automatic Event Detection

String Pattern	Logs Monitored
`ORA-31(13/37)`	Alert Log - DB
`ORA-603`	Alert Log – ASM
`ORA-00700`	Alert Log – ASM Proxy
`ORA-35(3|5|6)`	Alert Log – ASM IO Server
`ORA-40(20|36)`	
`ORA-403(0|1)`	
`ORA-2(27|39|40|55)`	
`ORA-1578`	
`ORA-2(5319|4982)`	
`ORA-56729`	
`OCI-31(06|35)`	
`ORA-445`	
`ORA-00600`	
`ORA-07445`	
`ORA-4(69|([7-8][0-9]|9([0-3]|[5-8])))`	
`ORA-297(01|02|03|08|09|10|40|70|71)`	
`ORA-3270(1|3|4)`	
`ORA-494`	
`System State dumped`	
`CRS-16(07|10|11|12)`	Alert Log - CRS

Related Topics:

Configuring Email Notification Details (page 4-11)

Purging the Repository Automatically (page 4-20)

4.1.3 Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections

Use Oracle Trace File Analyzer Collector to request an on-demand collection if you have determined a problem has occurred, and you are not able to resolve.

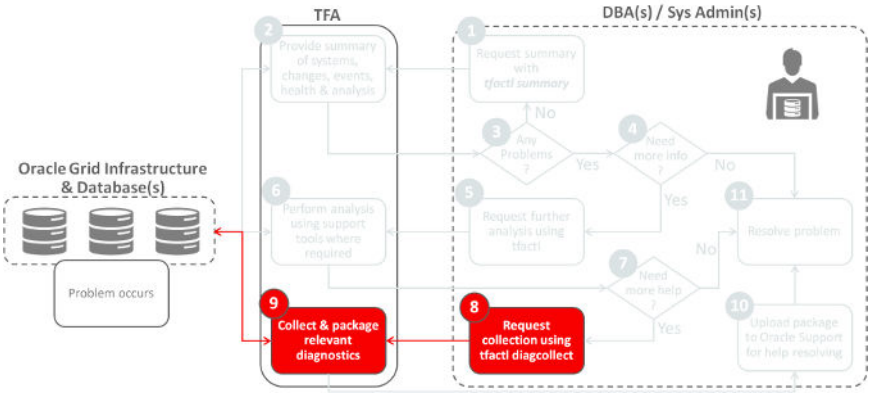
Provide the resulting collection to Oracle Customer Support to help you diagnose and resolve the problem.

Types of On-Demand Collections (page 4-5)
Oracle Trace File Analyzer Collector performs three types of on-demand collections. Use the `tfactl diagcollect` command to perform all on-demand collections.

4.1.3.1 Types of On-Demand Collections

Oracle Trace File Analyzer Collector performs three types of on-demand collections. Use the `tfactl diagcollect` command to perform all on-demand collections.

Figure 4-3 On-Demand Collections



- **Default Collections**

  Default collections gather all important files and diagnostics from all nodes for all components where the file has been updated within a particular time frame. Oracle Trace File Analyzer trims, if it deems the files to be excessive.

  The standard time period used for the default collections is the past 12 hours. However, you can adjust to any other time period.

- **Event Driven SRDC Collections**

  Event driven Service Request Data Collections (SRDC) gather all important files and diagnostics related to a particular event, such as an error.

  The files and diagnostics collected depend on the event the SRDC collection is for. Oracle Trace File Analyzer prompts you for any other important information that it needs. Providing information enables Oracle Trace File Analyzer to understand how best to collect diagnostics about each event.

- **Custom Collections**

  Custom collections allow you to provide granular control over exactly what, how, and from where collected.

Related Topics:

tfactl diagcollect (page F-17)

Running On-Demand Default Collections (page 4-22)

Running On-Demand Event-Driven SRDC Collections (page 4-24)

Running On-Demand Custom Collections (page 4-26)

4.2 Getting Started with Oracle Trace File Analyzer

This section introduces you to installing and configuring Oracle Trace File Analyzer.

Supported Platforms and Product Versions (page 4-6)
Review the supported platforms and product versions for Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector.

Oracle Grid Infrastructure Trace File Analyzer Installation (page 4-7)
Oracle Trace File Analyzer is automatically configured as part of the Oracle Grid Infrastructure configuration when running `root.sh` or `rootupgrade.sh`.

Oracle Database Trace File Analyzer Installation (page 4-8)
Oracle Trace File Analyzer is installed as part of the database installation.

Securing Access to Oracle Trace File Analyzer (page 4-9)
Running `tfactl` commands is restricted only to authorized users.

Masking Sensitive Data (page 4-10)
Configure Oracle Trace File Analyzer Collector to mask sensitive data in log files.

Configuring Email Notification Details (page 4-11)
Configure Oracle Trace File Analyzer to send an email when an automatic collection completes to the email address that is registered with Oracle Trace File Analyzer.

4.2.1 Supported Platforms and Product Versions

Review the supported platforms and product versions for Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector.

Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector are supported 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

Oracle Trace File Analyzer and Oracle Trace File Analyzer Collector are supported with Oracle Grid Infrastructure and/or Oracle Database versions 10.2 or later.

4.2.2 Oracle Grid Infrastructure Trace File Analyzer Installation

Oracle Trace File Analyzer is automatically configured as part of the Oracle Grid Infrastructure configuration when running `root.sh` or `rootupgrade.sh`.

Two `tfa` directories are created when Oracle Trace File Analyzer is installed as part of the Oracle Grid Infrastructure.

- **`Grid_home/tfa`:** This directory contains the Oracle Trace File Analyzer executables and some configuration files.
- **`ORACLE_BASE/tfa`:** Where `ORACLE_BASE` is the Oracle Grid Infrastructure owner's `ORACLE_BASE`. This directory contains the Oracle Trace File Analyzer metadata files and logs.

Note:

The `ORACLE_BASE` can be on a shared file system because Oracle Trace File Analyzer creates a node-specific directory under the `tfa` directory.

Oracle Trace File Analyzer uses the JRE version 1.8, which is shipped as part of the Oracle Grid Infrastructure 12.2 or Oracle Database 12.2 Home.

By default, Oracle Trace File Analyzer is configured to start automatically. The automatic start implementation is platform-dependent.

For example:

Linux systems: Automatic restart is accomplished by using,

`init`

or

An `init` replacement such as `upstart`

or

`systemd`

Microsoft Window:

Automatic restart is implemented as a Windows service.

Oracle Trace File Analyzer is not managed as one of the Cluster Ready Services (CRS) because it must be available if CRS is down.

- Start Oracle Trace File Analyzer as follows:

```
Grid_home/tfa/bin/tfactl start
```

For example:

```
$ /u01/app/12.2.0/grid/tfa/bin/tfactl start
Starting TFA..
start: Job is already running: oracle-tfa
Waiting up to 100 seconds for TFA to be started..
. . . . .
Successfully started TFA Process..
. . . . .
TFA Started and listening for commands
```

- Stop Oracle Trace File Analyzer as follows:

  ```
  Grid_home/tfa/bin/tfactl stop
  ```

  For example:

  ```
  $ /u01/app/12.2.0/grid/tfa/bin/tfactl stop
  Stopping TFA from the Command Line
  Stopped OSWatcher
  TFA is running  - Will wait 5 seconds (up to 3 times)
  TFA-00518 Oracle Trace File Analyzer (TFA) is not running (stopped)
  TFA Stopped Successfully
  . . .
  Successfully stopped TFA..
  ```

  Note:

  In the preceding example output, "`Stopped OSWatcher`" is seen only if you are using the download from My Oracle Support. Since `OSWatcher` is included only in the download and not in the Oracle Grid Infrastructure or Oracle Database install.

4.2.3 Oracle Database Trace File Analyzer Installation

Oracle Trace File Analyzer is installed as part of the database installation.

Oracle recommends that you run Oracle Trace File Analyzer in daemon mode, which is configured as `root` user.

1. To configure daemon mode as `root`:

   Either choose an appropriate option when running `root.sh` or `rootupgrade.sh`

   or

   Configure post-install by running the `tfa_home/install/roottfa.sh` script.

   When you choose this option, Oracle Trace File Analyzer is installed in the `ORACLE_BASE` of the current installation owner.

2. To use Oracle Trace File Analyzer in non-daemon mode, access it from `ORACLE_HOME/suptools/tfa/release/tfa_home` using:

   ```
   $ ORACLE_HOME/suptools/tfa/release/tfa_home/bin/tfactl command
   ```

When a user uses `tfactl` for the first time, Oracle Trace File Analyzer determines and creates the `TFA_BASE` directory structure. Oracle Trace File Analyzer maintains a configuration and trace file metadata database for every user who runs `tfactl`.

In non-daemon mode, the ability of a user to run the `tfactl` command determines Oracle Trace File Analyzer access control list. A user is able to use `tfactl`, if the user has operating system permissions to run `tfactl`. However, `tfactl` collects only the data the user has operating system permission to read.

When Oracle Trace File Analyzer is installed in daemon mode, the Oracle Trace File Analyzer daemon runs as `root`. The ability of the user to access Oracle Trace File Analyzer depends on that user being given specific access rights using the `tfactl access` command.

If a user has access right to Oracle Trace File Analyzer, then Oracle Trace File Analyzer collects any files from diagnostic directories in Oracle Trace File Analyzer that are marked as public. Specify the directory as private while adding to Oracle Trace File Analyzer, to restrict specific Oracle Trace File Analyzer users with sufficient permissions accessing it. Also, modify the settings using the `tfactl directory modify` command.

Related Topics:

tfactl access (page F-5)

tfactl print (page F-36)

tfactl directory (page F-21)

4.2.4 Securing Access to Oracle Trace File Analyzer

Running `tfactl` commands is restricted only to authorized users.

`tfactl` provides a command-line interface and shell in order to:

- Run any desired 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

To provision user access to tfactl:

1. To list the users who have access to `tfactl`:

   ```
   tfactl access lsusers
   ```

2. 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.

3. To remove a user from accessing `tfactl`:

   ```
   tfactl access remove -user user [-local]
   ```

4. To remove all users from accessing `tfactl`:

   ```
   tfactl access removeall [-local]
   ```

5. To reset user access to default:

   ```
   tfactl access reset
   ```

4.2.5 Masking Sensitive Data

Configure Oracle Trace File Analyzer Collector to mask sensitive data in log files.

Masking Sensitive Data is an optional feature wherein Oracle Trace File Analyzer Collector masks sensitive data in log files. Oracle Trace File Analyzer Collector 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 data redaction:

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 Collector automatically locates the `mask_strings.xml` files and starts replacing the sensitive data in the diagnostics it collects.

4.2.6 Configuring Email Notification Details

Configure Oracle Trace File Analyzer to send an email when an automatic collection completes to the email address that is registered with Oracle Trace File Analyzer.

Configure the system on which Oracle Trace File Analyzer is running to send emails. Otherwise, email notification feature does not work.

To configure email notification details:

1. To set 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 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. Inspect the referenced collection details to determine if you know the root cause.

   b. Resolve the underlying cause of the problem if you know how

   c. If you do not know the root cause of the problem, then log an SR with Oracle Support and upload the provided collection details

Related Topics:

Oracle Trace File Analyzer Collector Automated Diagnostic Collections (page 4-3)

4.3 Automatically Collecting Diagnostic Data Using the Oracle Trace File Analyzer Collector

Manage Oracle Trace File Analyzer Collector daemon, diagnostic collections, and the collection repository.

In addition, add hosts to the Oracle Trace File Analyzer Collector configuration, modify default communication ports, and configure SSL protocol.

Managing the Oracle Trace File Analyzer Daemon (page 4-12)
Oracle Trace File Analyzer Collector runs out of `init` on UNIX systems or `init/upstart/systemd` on Linux systems so that Oracle Trace File Analyzer Collector starts automatically whenever a node starts.

Viewing the Status and Configuration of Oracle Trace File Analyzer (page 4-12)
View the status of Oracle Trace File Analyzer across all the nodes in the cluster using either `tfactl print status` or `tfactl print config` commands.

Configuring the Host (page 4-14)
You must have `root` or `sudo` access to `tfactl` to add hosts to Oracle Trace File Analyzer configuration.

Configuring the Ports (page 4-15)
The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005. However, you can change if that port range is not available on your system.

Configuring SSL and SSL Certificates (page 4-15)
View and restrict SSL/TLS protocols. Configure Oracle Trace File Analyzer to use self-signed or CA-signed certificate.

Managing Collections (page 4-18)
Manage directories configured in Oracle Trace File Analyzer and diagnostic collections.

Managing the Repository (page 4-20)
All collections are stored in the Oracle Trace File Analyzer repository. The repository size is the maximum space Oracle Trace File Analyzer is able to use on disk to store collections.

4.3.1 Managing the Oracle Trace File Analyzer Daemon

Oracle Trace File Analyzer Collector runs out of `init` on UNIX systems or `init/upstart/systemd` on Linux systems so that Oracle Trace File Analyzer Collector 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 manually start or stop Oracle Trace File Analyzer:
   - `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.

4.3.2 Viewing the Status and Configuration of Oracle Trace File Analyzer

View the status of Oracle Trace File Analyzer across all the nodes in the cluster using either `tfactl print status` or `tfactl print config` commands.

To view the status and configuration settings of Oracle Trace File Analyzer:

1. To view the status of Oracle Trace File Analyzer all nodes in the cluster:

```
tfactl print status
```

For example:

```
$ tfactl print status
.-------------------------------------------------------------------------------
------------.
| Host  | Status of TFA | PID   | Port | Version    | Build ID             |
Inventory Status |
+-------+---------------+-------+------+------------+----------------------
+------------------+
| node1 | RUNNING       | 29591 | 5000 | 12.2.1.0.0 | 12210020160810105317 |
COMPLETE         |
| node2 | RUNNING       | 34738 | 5000 | 12.2.1.0.0 | 12210020160810105317 |
COMPLETE         |
'-------+---------------+-------+------+------------+----------------------
+------------------'
```

Displays the status of Oracle Trace File Analyzer across all nodes in the cluster, and also displays the Oracle Trace File Analyzer version and the port on which it is running.

2. To view configuration settings of Oracle Trace File Analyzer:

```
tfactl print config
```

For example:

```
$ 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 (page F-36)

4.3.3 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
      ```

      Oracle Trace File Analyzer displays the current node list it is aware of and prompts you to update this node list, if needed.

   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

4.3.4 Configuring the Ports

The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005. However, you can change if that port range is not available on your system.

When installed, 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
   ```

4.3.5 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 (page 4-16)
The Oracle Trace File Analyzer daemons in a cluster communicate securely using the SSL/TLS protocols.

Configuring Self-Signed Certificates (page 4-16)
Use `Java keytool` to replace self-signed SSL certificates with personal self-signed certificates.

Configuring CA-Signed Certificates (page 4-17)
Use `Java keytool` and `openssl` to replace self-signed SSL certificates with Certificate Authority (CA) signed certificates.

4.3.5.1 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
- TLCv1.1
- TLSv1

Oracle Trace File Analyzer always restricts use of older the protocols `SSLv3` and `SSLv2Hello`.

To view and restrict protocols:

1. To view the available and restricted protocols:

   ```
   tfactl print protocols
   ```

   For example:

   ```
   $ tfactl print protocols
   .--------------------------------------.
   |                node1                 |
   +--------------------------------------+
   | Protocols                            |
   +--------------------------------------+
   | Available : [TLSv1, TLSv1.2, TLSv1.1] |
   | Restricted : [SSLv3, SSLv2Hello]     |
   '--------------------------------------'
   ```

2. To restrict the use of certain protocols:

   ```
   tfactl restrictprotocol [-force] protocol
   ```

   For example:

   ```
   $ tfactl restrictprotocol TLSv1
   ```

4.3.5.2 Configuring Self-Signed Certificates

Use `Java keytool` to replace self-signed SSL certificates with personal self-signed certificates.

To configure Oracle Trace File Analyzer to use self-signed certificates:

1. Create a private key and keystore file containing the self-signed certificate for the server:

   ```
   $ keytool -genkey -alias server_full  -keyalg RSA -keysize 2048 -validity 18263 -
   keystore myserver.jks
   ```

2. Create a private key and keystore file containing the private key and self signed-certificate for the client:

```
$ keytool -genkey -alias client_full -keyalg RSA -keysize 2048 -validity 18263 -
keystore myclient.jks
```

3. Export the server public key certificate from the server keystore:

```
$ keytool -export -alias server_full -file myserver_pub.crt -keystore
myserver.jks -storepass password
```

4. Export the client public key certificate from the server keystore:

```
$ keytool -export -alias client_full -file myclient_pub.crt -keystore
myclient.jks -storepass password
```

5. Import the server public key certificate into the client keystore:

```
$ keytool -import -alias server_pub -file myserver_pub.crt -keystore myclient.jks
-storepass password
```

6. Import the client public key certificate into the server keystore:

```
$ keytool -import -alias client_pub -file myclient_pub.crt  -keystore
myserver.jks -storepass password
```

7. Lock down 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
```

4.3.5.3 Configuring CA-Signed Certificates

Use `Java keytool` and `openssl` to replace self-signed SSL certificates with 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 client and 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 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 Root CA 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. Lock down 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
```

4.3.6 Managing Collections

Manage directories configured in Oracle Trace File Analyzer and diagnostic collections.

Including Directories (page 4-19)
Add the directories to the Oracle Trace File Analyzer configuration to include the directories in diagnostic collections.

Managing the Size of Collections (page 4-19)
Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and the `diagcollection` option `-nocores` to reduce the size of collections.

4.3.6.1 Including Directories

Add the 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 (page F-21)

tfactl print (page F-36)

4.3.6.2 Managing the Size of Collections

Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and the `diagcollection` option `-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 (page F-17)

tfactl set (page F-3)

4.3.7 Managing the Repository

All collections are stored in the Oracle Trace File Analyzer 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 (page 4-20)

Purging the Repository Manually (page 4-21)

4.3.7.1 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 (`reposizeMB`)

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 reposizeMB
```

For example:

```
$ tfactl set reposizeMB=20480
```

Related Topics:

tfactl set (page F-3)

4.3.7.2 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:

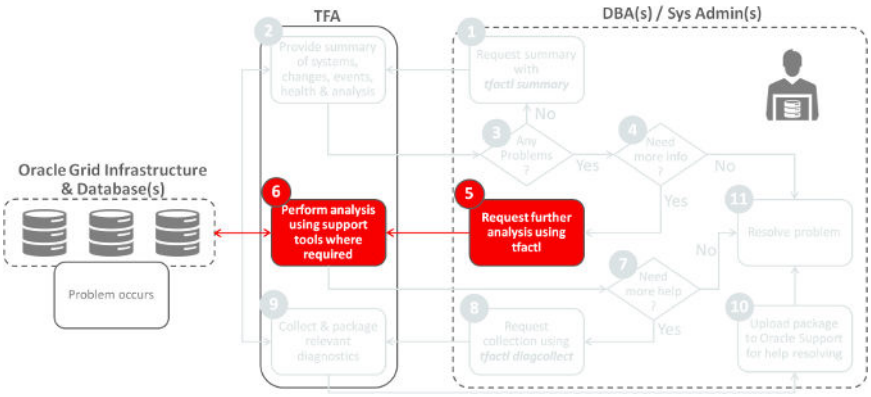
tfactl purge (page F-39)

tfactl print (page F-36)

4.4 Analyzing the Problems Identified

Use the `tfactl` command to perform further analysis against the database when you have identified a problem and you need more information.

Figure 4-4 Analysis



Related Topics:

tfactl analyze (page F-11)

Analyzing and Searching Recent Log Entries (page 4-32)

4.5 Manually Collecting Diagnostic Data

This section explains how to manually collect diagnostic data.

Running On-Demand Default Collections (page 4-22)
Use the `tfactl diagcollect` command to request a collection.

Running On-Demand Event-Driven SRDC Collections (page 4-24)
Use the `diagcollect -srdc` option to collect diagnostics needed for an Oracle Support Service Request Data Collection (SRDC).

Running On-Demand Custom Collections (page 4-26)
Use the custom collection options to collect diagnostic data from specific nodes, components, and directories.

4.5.1 Running On-Demand Default Collections

Use the `tfactl diagcollect` command to request a collection.

Oracle Trace File Analyzer stores all collections in the repository directory of the Oracle Trace File Analyzer installation.

The standard time period used for the default collections is the past 12 hours. However, you can adjust to any other time period.

To run on-demand default collections:

1. To request default collection:

```
tfactl diagcollect
```

For example:

```
$ 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
```

Adjusting the Time Period for a Collection (page 4-23)
By default, `diagcollect` trims and collects all important log files, from all nodes, for all components, where the file has been updated in the past 12 hours.

Related Topics:

tfactl diagcollect (page F-17)

Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections (page 4-5)

4.5.1.1 Adjusting the Time Period for a Collection

By default, `diagcollect` trims and collects all important log files, from all nodes, for all components, where the file has been updated in the past 12 hours.

Narrow down the problem further and collect the minimal possible data.

There are four different ways of specifying a time period for the collection.

Use whichever is most appropriate in your situation based on what you know about when the symptoms of the problem occurred and any anything relevant that might have contributed to it.

Table 4-2 Adjusting the Time Period for a Collection

Command	Description
`-since nh|d`	Collect since the previous n hours or days.
`-from "yyyy-mm-dd"`	Collect from the date and optionally time specified. Valid date / time formats: • `"Mon/dd/yyyy hh:mm:ss"` • `"yyyy-mm-dd hh:mm:ss"` • `"yyyy-mm-ddThh:mm:ss"` • `"yyyy-mm-dd"`
`-to "yyyy-mm-dd"`	Collect to the date and optionally time specified. Valid date / time formats: • `"Mon/dd/yyyy hh:mm:ss"` • `"yyyy-mm-dd hh:mm:ss"` • `"yyyy-mm-ddThh:mm:ss"` • `"yyyy-mm-dd"`
`-for "yyyy-mm-dd"`	Collect for the specified date. Valid date / time formats: • `"Mon/dd/yyyy"` • `"yyyy-mm-dd"`

To adjust the time period for a collection:

1. To adjust the time period:

   ```
   tfactl diagcollect -since nh|d
   ```

   For example:

   To do a collection covering the past 2 hours:

   ```
   $ tfactl diagcollect -since 2h
   ```

   To do a collection covering the past 3 days:

   ```
   $ tfactl diagcollect -since 3d
   ```

   To do a collection for a specific date:

   ```
   $ tfactl diagcollect -for "2016-08-15"
   ```

   To do a collection from one particular date to another:

   ```
   $ tfactl diagcollect -from "2016-08-15" -to "2016-08-17"
   ```

Related Topics:

tfactl diagcollect (page F-17)

4.5.2 Running On-Demand Event-Driven SRDC Collections

Use the `diagcollect -srdc` option to collect diagnostics needed for an Oracle Support Service Request Data Collection (SRDC).

Event-driven SDRC collections require components from the Oracle Trace File Analyzer Database Support Tools Bundle.

Download Oracle Trace File Analyzer Database Support Tools Bundle from My Oracle Support Note 1513912.2:

https://support.oracle.com/rs?type=doc&id=1513912.2

To run event-driven SRDC collections:

1. To run event-driven SRDC collections:

   ```
   tfactl diagcollect -srdc srdc_type
   ```

2. To obtain a list of different types of SRDC collections:

   ```
   tfactl diagcollect -srdc -help
   ```

   For example:

   ```
   $ 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 |
   '------------+-----------+-------+------'
   ```

3. Use the same tagging, naming, and time arguments with the SRDC collections as with other collections:

```
Usage : tfactl diagcollect -srdc srdc_profile [-tag description] [-z filename] [-
since nh|d| -from time -to time | -for time]
```

Related Topics:

tfactl diagcollect (page F-17)

Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections (page 4-5)

4.5.3 Running On-Demand Custom Collections

Use the custom collection options to collect diagnostic data from specific nodes, components, and directories.

By default, Oracle Trace File Analyzer:

- Collects from all nodes in the cluster
- Collects from all Oracle database and Oracle Grid Infrastructure components
- Compresses the collections into the repository directory in a `zip` file with the following format:

  repository/collection_date_time/node_all/node.tfa_date_time.zip
- Copies back all zip files from remote notes to the initiating node
- Trims the files around the relevant time
- Includes any relevant core files it finds

Also, Oracle Trace File Analyzer Collector collects files from any other directories you want.

Collecting from Specific Nodes (page 4-27)

Collecting from Specific Components (page 4-27)

Collecting from Specific Directories (page 4-28)

Changing the Collection Name (page 4-29)

Preventing Copying Zip Files and Trimming Files (page 4-30)

Performing Silent Collection (page 4-30)

Preventing Collecting Core Files (page 4-30)

Collecting Incident Packaging Service Packages (page 4-31)

Related Topics:

tfactl diagcollect (page F-17)

tfactl ips (page F-23)

Oracle Trace File Analyzer Collector On-Demand Diagnostic Collections (page 4-5)

4.5.3.1 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 -since 1d -node myserver65
   ```

Related Topics:

tfactl diagcollect (page F-17)

4.5.3.2 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 -since 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 -since 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 4-3 Component Options

Component Option	Description
`-database database names`	Collects database logs from databases specified in a comma-separated list.
`-asm`	Collects Oracle ASM logs.
`-crsclient`	Collects Client Logs that are under `GIBASE/diag/clients`.
`-dbclient`	Collects Client Logs that are under `DB ORABASE/diag/clients`.
`-dbwlm`	Collects DBWLM logs.

Table 4-3 (Cont.) Component Options

Component Option	Description
`-tns`	Collects TNS logs.
`-rhp`	Collects RHP logs.
`-procinfo`	Collects `Gathers stack` and `fd` from `/proc` 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 `/var/log/messages`.
`-ashhtml`	Collects Generate ASH HTML Report.
`-ashtext`	Collects Generate ASH TEXT Report.
`-awrhtml`	Collects AWRHTML logs.

Related Topics:

tfactl diagcollect (page F-17)

4.5.3.3 Collecting from Specific Directories

To collect from specific directories:

1. To include all files, no matter the type or time last updated, from other directories in the collection:

   ```
   tfactl diagcollect -collectdir 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 -collectdir /tmp_dir1,/tmpdir_2
   ```

2. To collect from all directories:

```
tfactl diagcollect -collectalldirs
```

Oracle Trace File Analyzer collects from all files in the directory irrespective of time or time range.

For example:

To collect all standard trace and diagnostic files updated in the past day, plus all files from any `collectall` directories, no matter when they were updated:

```
$ tfactl diagcollect -since 1d -collectalldirs
```

Related Topics:

tfactl diagcollect (page F-17)

4.5.3.4 Changing the Collection Name

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 -since 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 -since 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 (page F-17)

4.5.3.5 Preventing Copying Zip Files and Trimming Files

By default, Oracle Trace File Analyzer Collector:

- Copies back all zip files from remote notes to the initiating node
- Trims the 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 -since 1d -nocopy
   ```

2. To avoid trimming files:

   ```
   -notrim
   ```

   For example:

   ```
   $ tfactl diagcollect -since 1d -notrim
   ```

Related Topics:

tfactl diagcollect (page F-17)

4.5.3.6 Performing Silent Collection

1. To initiate a silent collection:

   ```
   -silent
   ```

   The `diagcollect` command is submitted as a background process.

   For example:

   ```
   $ tfactl diagcollect -since 1d -silent
   ```

Related Topics:

tfactl diagcollect (page F-17)

4.5.3.7 Preventing Collecting Core Files

1. To prevent core files being included:

   ```
   -nocores
   ```

   For example:

   ```
   $ tfactl diagcollect -since 1d -nocores
   ```

Related Topics:

tfactl diagcollect (page F-17)
Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

4.5.3.8 Collecting Incident Packaging Service Packages

Oracle Trace File Analyzer is capable of calling the Incident Packaging Service (IPS), which collects files from the Automatic Diagnostic Repository (ADR).

1. To run Incident Packaging Service:

   ```
   $ tfactl ips
   ```

2. To collect with Incident Packaging Service:

   ```
   $ tfactl diagcollect -ips
   ```

3. To show all Incident Packaging Service incidents:

   ```
   $ tfactl ips show incidents
   ```

4. To show all Incident Packaging Service problems:

   ```
   $ tfactl ips show problems
   ```

5. To show all Incident Packaging Service problems:

   ```
   $ tfactl ips show package
   ```

6. To see available `diagcollect` Incident Packaging Service options:

   ```
   $ tfactl diagcollect -ips -h
   ```

7. To run Incident Packaging Service collection interactively:

   ```
   $ tfactl diagcollect -ips
   ```

   When you run interactively, you are prompted to select the Automatic Diagnostic Repository area to collect from.

8. To run Incident Packaging Service collection in silent mode:

   ```
   $ tfactl diagcollect -ips -adrbasepath path -adrhomepath path
   ```

9. Use the standard `diagcollect` options to limit the scope of Incident Packaging Service collection.

   For example, to collect Incident Packaging Service packages for the given ADR `basepath/homepath` in the last hour in the local node:

   ```
   $ tfactl diagcollect -ips -adrbasepath /scratch/app/oragrid -adrhomepath diag/crs/
   hostname/crs -since 1h -node local
   ```

10. To collect Automatic Diagnostic Repository details about a specific incident ID:

    ```
    $ tfactl diagcollect -ips -incident incident id -node local
    ```

11. To collect Automatic Diagnostic Repository details about a specific problem ID:

    ```
    $ tfactl diagcollect -ips -problem problem id -node local
    ```

To change the contents of the Incident Packaging Service package, you can initiate collection, pause it, manipulate the package, and then resume collection.

12. To collect Automatic Diagnostic Repository details about a specific incident id on the local node and pause for Incident Packaging Service package manipulation:

```
$ tfactl diagcollect -ips -incident incident id -manageips -node local
```

13. To print all paused Oracle Trace File Analyzer Incident Packaging Service collections:

```
$ tfactl print suspendedips
```

14. To resume a suspended Oracle Trace File Analyzer Incident Packaging Service collection:

```
$ tfactl diagcollect -resumeips collection id
```

Related Topics:

tfactl diagcollect (page F-17)

tfactl ips (page F-23)

tfactl print (page F-36)

4.6 Analyzing and Searching Recent Log Entries

Use the `tfactl analyze` command to analyze and search recent log entries.

To analyze and search recent log entries:

1. To analyze all important recent log entries:

```
tfactl analyze -since n[h|d]
```

Specify the period of time to analyze in either hours or days.

For example:

```
tfactl analyze -since 14d
```

The command output shows you a summary of errors found in the logs during the period specified.

2. To search for all occurrences of a particular message or error code over a specified period of hours or days:

```
tfactl analyze -search "message" -since n[h|d]
```

For example:

```
$ tfactl analyze -search "ORA-006" -since 14d
```

Related Topics:

tfactl analyze (page F-11)

Analyzing the Problems Identified (page 4-21)

4.7 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.

4.7.1 Managing Automatic Diagnostic Repository Log and Trace Files

Use the `managelogs` command to manage Automatic Diagnostic Repository log and trace files.

The `-purge` command removes files managed by Automatic Diagnostic Repository (Files are cleared from "`ALERT`", "`INCIDENT`", "`TRACE`", "`CDUMP`", "`HM`", "`UTSCDMP`", "`LOG`" under diagnostic destinations) and provides details about the change in the file system space.

For diagnostic destinations where there are large numbers of files the command might take a while. Check the removal in progress from corresponding directories.

You must have operating system privilege over corresponding diagnostic destinations to remove the files.

To manage Automatic Diagnostic Repository log and trace files:

1. To limit purge or show operations to only files older than a specified time:

   ```
   $ tfactl managelogs -older nm|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. Perform a dry run to get an estimate of how many files are removed and how much space is freed by executing the `purge` command with 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 (page F-39)

4.7.2 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
```

4.7.3 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, Oracle Trace File Analyzer runs an automatic purge every 60 minutes of 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
```

4.8 Upgrading Oracle Trace File Analyzer Collector by Applying a Patch Set Update

Always upgrade to the latest version whenever possible to include bug fixes, new features, and optimizations.

Applying the patch set update automatically updates Oracle Trace File Analyzer. The latest version of Oracle Trace File Analyzer is shipped with each new database and Oracle Grid Infrastructure patch set update. The patch set update version is normally three months behind the version that is released on My Oracle Support.

When a new patch set update is applied to Oracle Grid Infrastructure home or database home, Oracle Trace File Analyzer upgrades automatically if the version in the PSU is greater than the version that is currently installed.

The latest Oracle Trace File Analyzer version is available on My Oracle Support Note 1513912.2, three months before it is available in a patch set update.

https://support.oracle.com/rs?type=doc&id=1513912.2

When updating Oracle Trace File Analyzer through patch set update, Oracle Trace File Analyzer Database Support Tools Bundle is not updated automatically. Download and update the support tools from My Oracle Support Note 1513912.2.

4.9 Troubleshooting Oracle Trace File Analyzer

Enable specific trace levels when reproducing a problem to obtain sufficient diagnostics.

To quickly enable or disable the correct trace levels, use the `dbglevel` option.

All the required trace level settings are organized into problem-specific trace profiles.

To set trace levels:

1. To set a trace profile:

```
tfactl dgblevel -set profile
```

5

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 is estimated 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.

Managing the Oracle Cluster Health Advisor Repository (page 5-8)
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 (page 5-9)
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:

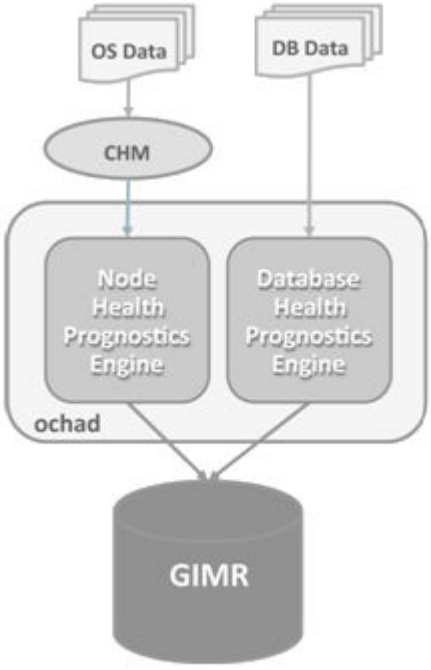
Introduction to Oracle Cluster Health Advisor (page 1-6)

5.1 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 5-1 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.

5.2 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.

5.3 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 5-1 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 *`oltpacbd`*.

```
$ 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 insufficent
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 5-2 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 (page E-7)

5.4 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 5-2 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%     0.00%
```

Related Topics:

chactl calibrate (page E-5)

chactl query calibration (page E-10)

chactl Command Reference (page E-1)

5.5 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.

5.6 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
```

5.7 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 the 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.
```

6

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.

Related Topics:

Introduction to Memory Guard (page 1-7)

6.1 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.

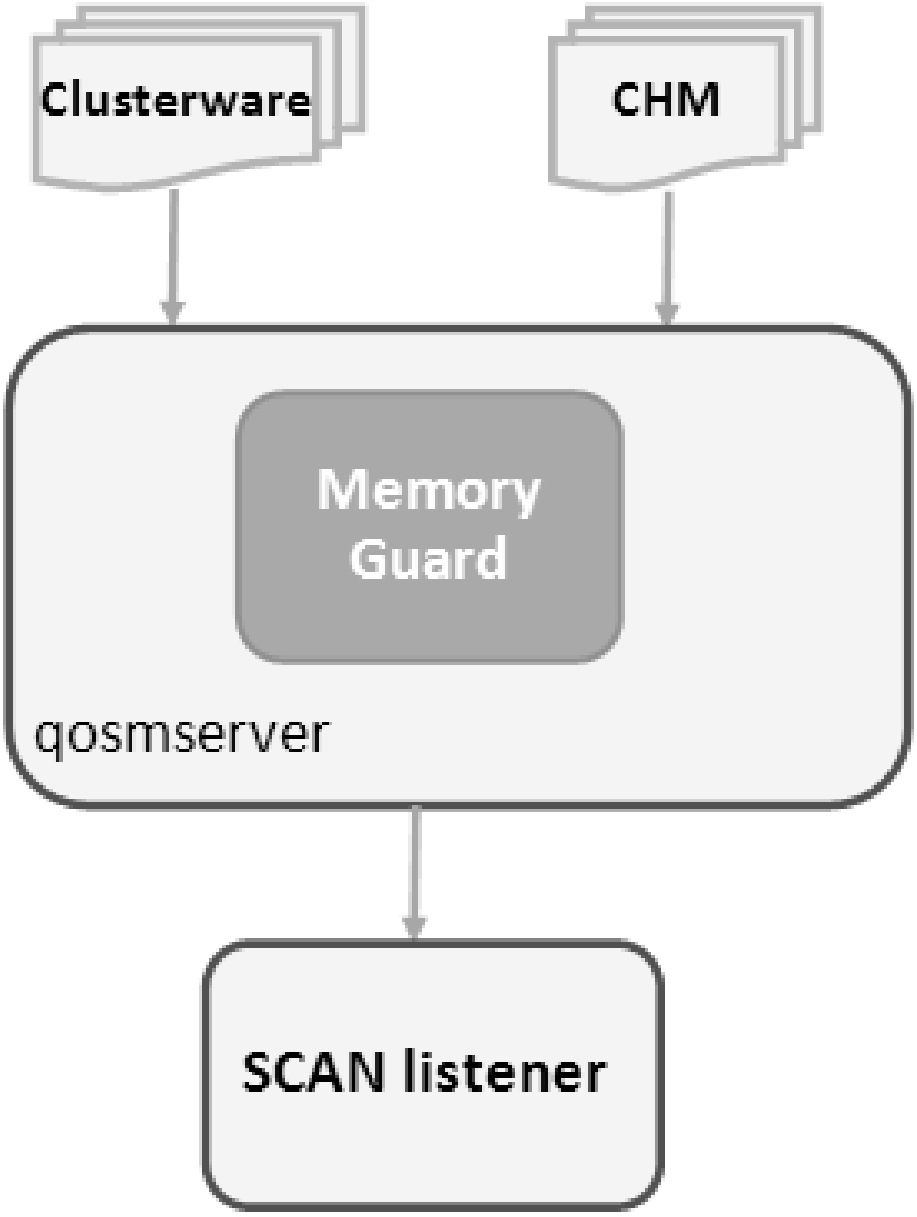
Running out of memory can result in failed transactions or, in extreme cases, a restart of the node resulting in the loss of availability and resources for your applications.

6.2 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 6-1 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.

6.3 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 qosmserver
```

Example 6-1 Verifying that Memory Guard is Running on a Node

The following example shows sample output of the status of Memory Guard on `qosmserver`.

```
$ srvctl status qosmserver
QoS Management Server is enabled.
QoS Management Server is running on node nodeABC
```

6.4 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 6-2 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>
```

7

Resolving Database and Database Instance Hangs

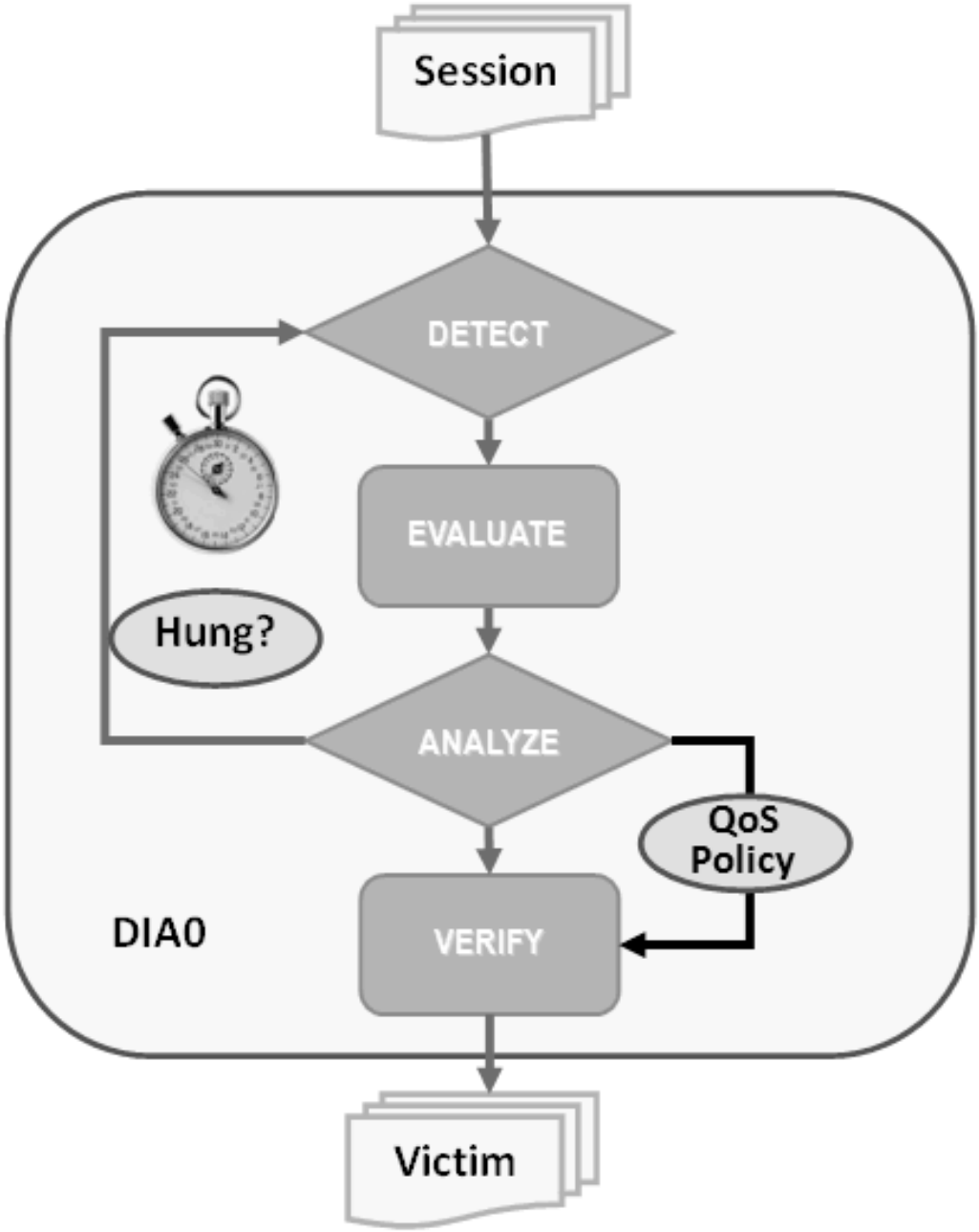
Hang Manager preserves the database performance by resolving hangs and keeping the resources available.

Related Topics:

7.1 Hang Manager Architecture

Hang Manager autonomously runs as a `DIA0` task within the database.

Figure 7-1 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.

7.2 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.

7.3 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 7-1 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/hm11/incident/incdir_111/
hm11_dia0_11111_i111.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.

kjznshngtbldmp: Hang's QoS Policy and Multiplier Checksum 0x0

  Inst  Sess   Ser             Proc  Wait
   Num    ID    Num      OSPID  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 7-2 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/hm1/hm1/trace/hm1_dia0_i1111.trc
(incident=1111):
```

```
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hm1/hm1/incident/incdir_1111/
hm1_dia0_11111_i1111.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 7-3 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/hm1/hm11/trace/hm11_dia0_11111.trc
(incident=169):
ORA-32701: Possible hangs up to hang ID=1 detected
Incident details in: .../oracle/log/diag/rdbms/hm1/hm11/incident/incdir_169/
hm11_dia0_30676_i169.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.
```

8

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.

Related Topics:

Managing the Cluster Resource Activity Log (page D-1)

crsctl query calog (page D-1)

crsctl get calog maxsize (page D-8)

crsctl get calog retentiontime (page D-9)

crsctl set calog maxsize (page D-9)

crsctl set calog retentiontime (page D-10)

8.1 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.

See Also:

Oracle Database 2 Day + Real Application Clusters Guide

8.2 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 (page B-1)

8.3 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:

9

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.

This chapter contains the following sections:

Related Topics:

Introduction to Oracle Database Quality of Service (QoS) Management (page 1-8)

9.1 What Does Oracle Database Quality of Service (QoS) Management Manage?

Oracle Database Quality of Service (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 various applications.

Oracle Database QoS Management manages the CPU resource for a cluster. Oracle Database QoS Management does not manage I/O resources. Therefore, Oracle Database QoS Management does not effectively manage I/O intensive applications. 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 altering the consumer group mappings in the resource plan used by Oracle Database Resource Manager.

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

---

9.2 How Does Oracle Database Quality of Service (QoS) Management Work?

Oracle Database Quality of Service (QoS) Management uses a resource management plan and user-specific performance objectives to allocate resources to defined workloads.

With Oracle Database, 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 Oracle Database QoS Management 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.
5. Provides an analysis of the predicted impact to performance levels for each Performance Class if that recommendation is implemented.
6. Implements the actions listed in the recommendation when directed to by the Oracle Database QoS Management administrator.
7. Evaluates the system to verify that each Performance Class is meeting its Performance Objective after the resources have been reallocated.

9.3 Overview of Metrics

Oracle Database Quality of Service (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 various 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.

Oracle Database QoS Management uses a standardized set of metrics, which are collected by all the servers in the system. 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. Since 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.

See Also:

"*Oracle Database Quality of Service Management User's Guide*"

9.4 Benefits of Using Oracle Database Quality of Service (QoS) Management

Oracle Database QoS Management helps manage the resources shared by databases and their services in a cluster.

In a typical company, when the response times of your applications are not within acceptable levels, problem resolution can be 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 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 required 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 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. Instead, Oracle Database QoS Management works to remove obstacles that prevent your applications from running at their optimal performance levels.

A

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]
[-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]
[-daemon [-id ID] -set parameter | [-id ID] -unset parameter | all | [-id ID] -get
parameter | all]
AUTORUN_SCHEDULE=value | AUTORUN_FLAGS=flags | NOTIFICATION_EMAIL=email |
PASSWORD_CHECK_INTERVAL=number of hours | collection_retention=number of days
[-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 | -idmdbruntime]
[-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]
[-checkfailedуploads]
[-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 | -
initpresetup | -h]
[-d start|start_debug|stop|status|info|stop_client|nextautorun|-h]
[-daemon [-id ID] -set parameter | [-id ID] -unset parameter | all | [-id ID] -get
parameter | all]
AUTORUN_SCHEDULE=value > | AUTORUN_FLAGS=flags | NOTIFICATION_EMAIL=email |
PASSWORD_CHECK_INTERVAL=number of hours | collection_retention=number of days
[-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 | 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 | 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]
```


A.1 Running Generic Oracle ORAchk and Oracle EXAchk Commands

List of command options common to Oracle ORAchk and Oracle EXAchk.

Syntax

```
[-a]
[-v]
[-debug]
```

```
[-daemon]
[-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 A-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.
`-daemon`	Runs only if the daemon is running.
`-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=`*`comma-delimited list of email addresses`*`"`	Sends a test email to validate email configuration.
`-sendemail “NOTIFICATION_EMAIL=`*`comma-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.
`-dbparallel [`*`n`*`]`	Runs the `SQL`, `SQL_COLLECT`, and `OS` health checks in parallel, using *n* number of child processes. Default is 25% of CPUs.

Table A-1 (Cont.) Generic Commands

Option	Description
`-dbparallelmax`	Runs the `SQL`, `SQL_COLLECT`, and `OS` health checks in parallel, using the maximum number of child processes.

A.2 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]
[-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 A-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` *nodes*	Specify a comma-delimited list of node names to run only on a subset of nodes.
`-dbnames` *db_names*	Specify a comma-delimited list of database names to run only on a subset of databases.

Table A-2 (Cont.) Scope of Checks

Command	Description
`-dbnone`	Does not prompt for database selection and skips all the database checks.
`-dball`	Does not prompt for database selection and runs the database checks on all databases discovered on the system.
`-localonly`	Runs only on the local node.
`-cells` *`cells`*	Specify a comma-delimited list of storage server names to run the checks only on a subset of storage servers.
`-ibswitches` *`switches`*	Specify a comma-delimited list of InfiniBand switch names to run the checks only on a subset of InfiniBand switches.
`-profile` *`profile`*	Specify a comma-delimited list of profiles to run only the checks in the specified profiles.
`-excludeprofile` *`profile`*	Specify a comma-delimited list of profiles to exclude the checks in the specified profiles.
`-check` *`check_id`*	Specify a comma-delimited list of check IDs to run only the checks specified in the list check IDs.
`-excludecheck` *`check_id`*	Specify a comma-delimited list of check IDs to exclude the checks specified in the list of check IDs.
`-skip_usr_def_checks`	Does not run the checks specified in the user-defined `xml` file.

A.3 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 A-3 Managing Output

Option	Description
`-syslog`	Writes JSON results to syslog.

Table A-3 (Cont.) Managing Output

Option	Description
`-tag tagname`	Appends the *tagname* specified to the output report name. The *tagname* must contain only alphanumeric characters.
`-o`	Argument to an option. If `-o` is followed by `v`, (or `verbose`, 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 old_report new_report [-outfile output_HTML]`	Reports the difference between the two HTML reports. Specify a directory name or a ZIP file or an HTML report file as *old_report* and *new_report*.
`-merge [-force] collections`	Merges a comma-delimited list of collections and prepares a single report.

A.4 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 A-4 Uploading Results to Database

Option	Description
`-setdbupload all|variable_names`	Sets the values in the wallet to upload health check run results to the database. `all`: Sets all the variables in the wallet. *variable_names*: Specify a comma-delimited list of variables to set.
`-unsetdbupload all|variable_names`	Unsets the values in the wallet to upload health check run results to the database. `all`: Unsets all the variables in the wallet. *variable_names*: Specify a comma-delimited list of variables to unset.

Table A-4 (Cont.) Uploading Results to Database

Option	Description
`-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.
`-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 filed collection uploads. *list of failed collections*: Specify a comma-delimited list of collections to upload.

A.5 Configuring the Daemon Mode

Use the daemon to configure automatic health check runs at scheduled intervals.

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 as described in "Automating the Daemon Mode Operations".

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.

A.6 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]
[-initpresetup]
```

Parameters

Table A-5 Daemon Options

Option	Description
`[-id id] –set daemon_option`	Optionally use `id` with the `set` command to set specific daemon usage profiles.
`[-id id] -unset daemon_option | all`	Unsets the parameter. Use with `–id id` 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 `–id id` 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.

Table A-5 (Cont.) Daemon Options

Option	Description
`-initpresetup`	Sets the `root` user equivalency for `COMPUTE`, `STORAGE`, and `IBSWITCHES` (`root` equivalency for `COMPUTE` nodes is mandatory for setting up auto restart functionality).

A.7 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 A-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 directories`	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`

Table A-6 (Cont.) List of Oracle ORAchk and Oracle EXAchk File Attribute Tracking Options

Option	Description
`-fileattr check -baseline` *`baseline snapshot path`*	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`

B

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.

B.1 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 B-1 oclumon debug Command Parameters

Parameter	Description

`log` *daemon* `module:` *log_level*	Use this option change the log level of daemons and daemon modules.
	Supported daemons are:
	`osysmond`
	`ologgerd`
	`client`
	`all`
	Supported daemon modules are:
	`osysmond`: `CRFMOND`, `CRFM`, and `allcomp`
	`ologgerd`: `CRFLOGD`, `CRFLDREP`, `CRFM`, and `allcomp`
	`client`: `OCLUMON`, `CRFM`, and `allcomp`
	`all`: `allcomp`
	Supported *log_level* values are `0`, `1`, `2`, and `3`.

`version`	Use this option to display the versions of the daemons.

Example B-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
```

B.2 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 B-2 oclumon dumpnodeview Command Parameters

Parameter	Description
`-allnodes`	Use this option to dump the node views of all the nodes in the cluster.
`-n node1 node2`	Specify one node or several nodes in a space-delimited list for which you want to dump the node view.
`-last "duration"`	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 "time_stamp" -e "time_stamp"`	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.

Table B-2 (Cont.) oclumon dumpnodeview Command Parameters

Parameter	Description
`-i interval`	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 "format type"`	Specify the output format. "*format type*" can be `legacy`, `tabular`, or `csv`. The default format is mostly tabular with legacy for node view parts with only one row.
`-dir directory`	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 dir_name` 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 dir_name -append`, appends the data of the current run with the previous one in the specified directory.
`-procag`	Outputs the process of the node view, aggregated by category: • DBBG (DB backgrounds) • DBFG (DB foregrounds) • CLUST (Cluster) • OTHER (other processes)
`-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 B-3 oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
`#pcpus`	The number of physical CPUs.
`#vcpus`	Number of logical compute units.
`cpuht`	CPU hyperthreading enabled (Y) or disabled (N).
`chipname`	The name of the CPU vendor.
`cpu`	Average CPU utilization per processing unit within the current sample interval (%).
`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).
`mcache`	Amount of physical RAM used for file buffers plus the amount of physical RAM used as cache memory (KB). On 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.

Table B-3 (Cont.) oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
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.
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.
rtprocs	Number of real-time processes.
rtprocsoncpu	The current number of real-time processes running on the CPU.

Table B-3 (Cont.) oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
#fds	Number of open file descriptors. *or* Number of open handles on Microsoft Windows.
#sysfdlimit	System limit on 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 B-4 oclumon dumpnodeview PROCESSES View Metric Descriptions

Metric	Description
name	The name of the process executable.
pid	The process identifier assigned by the operating system.
#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.
privmem	Process private memory usage (KB).
shm	Process shared memory usage (KB). **Note:** This metric is not available on Microsoft Windows, Solaris, and AIX 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. *or* Number of open handles by this process on Microsoft Windows.

Table B-4 (Cont.) oclumon dumpnodeview PROCESSES View Metric Descriptions

Metric	Description
`#threads`	Number of threads created by this process.
`priority`	The process priority.
`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 B-5 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 `WAIT` 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: • `SWAP` • `SYS` • `OCR` • `ASM` • `VOTING`

Table B-6 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).

Table B-6 (Cont.) oclumon dumpnodeview NICS View Metric Descriptions

Metric	Description
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).
errsin	Average error rate for incoming packets within the current sample interval (errors per second).
errsout	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 B-7 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.

Table B-7 (Cont.) oclumon dumpnodeview FILESYSTEMS View Metric Descriptions

Metric	Description
`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 B-8 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.
`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.

Table B-8 (Cont.) oclumon dumpnodeview PROTOCOL ERRORS View Metric Descriptions

Metric	Description
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 B-9 oclumon dumpnodeview CPUS View Metric Descriptions

Metric	Description
cpu*id*	Virtual CPU.
sys-*usage*	CPU usage in system space.
user-*usage*	CPU usage in user space.
nice	Value of NIC for a specific CPU.
usage	CPU usage for a specific CPU.
iowait	CPU wait time for I/O operations.

Example B-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 B-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 B-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 B-5 Node View Output

```
----------------------------------------
Node: rwsak10 Clock: '2016-05-08 02.11.25-0800' SerialNo:155631
----------------------------------------

SYSTEM:
#pcpus: 2 #vcpus: 24 cpuht: Y chipname: Intel(R) cpu: 1.23 cpuq: 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 procsoncpu: 6 rtprocs: 62 rtprocsoncpu 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
sda1 ior: 0.000 iow: 61.495 ios: 629 qlen: 0 wait: 0 type: SYS
.
.
.

NICS:

lo netrr: 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 netrr: 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
```

B.3 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 B-10 oclumon manage Command Parameters

Parameter	Description
`-repos {{changeretentiontime` *`time`*`} | {changerepossize` *`memory_size`*`}}`	The `-repos` flag is required to specify the following Cluster Health Monitor repository-related options: • `changeretentiontime` *`time`*: 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 *does not* change retention time. • `changerepossize` *`memory_size`*: 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.
`-get` *`key1`* `[`*`key2`* `...]`	Use this option to obtain Cluster Health Monitor repository information using the following keywords: `repsize`: Size of the Cluster Health Monitor repository, in seconds `reppath`: Directory path to the Cluster Health Monitor repository `master`: Name of the master node `alllogger`: Special key to obtain a list of all nodes running Cluster Logger Service `mylogger`: Special key to obtain the node running the Cluster Logger Service which is serving the current node • `-details`: Use this option with `alllogger` and `mylogger` for listing nodes served by the Cluster Logger Service You can specify any number of keywords in a space-delimited list following the `-get` flag.
`-h`	Displays online help for the `oclumon manage` 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 B-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 changerepossize 6000
```

B.4 oclumon version

Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.

Syntax

```
oclumon version
```

Example B-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.
```

C

Diagnostics Collection Script

Running diagnostics collection script provide additional information so My Oracle Support can resolve problems.

Syntax

```
diagcollection.pl {--collect [--crs | --acfs | -all] [--chmos [--incidenttime time
[--incidentduration time]]] [--adr location [--aftertime time [--beforetime time]]]
[--crshome path | --clean | --coreanalyze}]
```

Note:

Prefix `diagcollection.pl` script arguments with two dashes (`--`).

Parameters

Table C-1 diagcollection.pl Script Parameters

Parameter	Description
`--collect`	Use this parameter with any of the following arguments: • `--crs`: Use this argument to collect Oracle Clusterware diagnostic information • `--acfs`: Use this argument to collect Oracle ACFS diagnostic information **Note:** You can only use this argument on UNIX systems. • `--all`: (default) Use this argument to collect all diagnostic information except Cluster Health Monitor (OS) data. • `--chmos`: Use this argument to collect the following Cluster Health Monitor diagnostic information `--incidenttime` *`time`*: Use this argument to collect Cluster Health Monitor (OS) data from the specified time **Note:** The time format is `MM/DD/YYYYHH24:MM:SS`. `--incidentduration` *`time`*: Use this argument with `--incidenttime` to collect Cluster Health Monitor (OS) data for the duration after the specified time **Note:** The time format is `HH:MM`. If you do not use `--incidentduration`, then all Cluster Health Monitor (OS) data after the time you specify in `--incidenttime` is collected. • `--adr` *`location`*: The Automatic Diagnostic Repository Command Interpreter (ADRCI) uses this argument to specify a location in which to collect diagnostic information for ADR **See Also:** *Oracle Database Utilities* for more information about ADRCI • `--aftertime` *`time`*: Use this argument with the `--adr` argument to collect archives after the specified time **Note:** The time format is `YYYYMMDDHHMISS24`. • `--beforetime` *`time`*: Use this argument with the `--adr` argument to collect archives before the specified time **Note:** The time format is `YYYYMMDDHHMISS24`. • `--crshome` *`path`*: Use this argument to override the location of the Oracle Clusterware home **Note:** The `diagcollection.pl` script typically derives the location of the Oracle Clusterware home from the system configuration (either the `olr.loc` file or the Microsoft Windows registry), so this argument is not required.
`--clean`	Use this parameter to clean up the diagnostic information gathered by the `diagcollection.pl` script. **Note:** You cannot use this parameter with `--collect`.

Table C-1 (Cont.) diagcollection.pl Script Parameters

Parameter	Description
`--coreanalyze`	Use this parameter to extract information from core files and store it in a text file. **Note:** You can only use this parameter on UNIX systems.

D

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:

D.1 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 D-1 crsctl query calog Command Parameters

Parameter	Description
`-aftertime "timestamp"`	Displays the activities logged after a specific time. Specify the timestamp in the `YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM]` or `YYYY-MM-DD` or `HH24:MI:SS[.FF][TZH:TZM]` format. `TZH` and `TZM` stands for time zone hour and minute, and `FF` stands for microseconds. If you specify `[TZH:TZM]`, then the `crsctl` command assumes UTC as time zone. If you do not specify `[TZH:TZM]`, then the `crsctl` command assumes the local time zone of the cluster node from where the `crsctl` command is run. Use this parameter with `-beforetime` to query the activities logged at a specific time interval.
`-beforetime "timestamp"`	Displays the activities logged before a specific time. Specify the timestamp in the `YYYY-MM-DD HH24:MI:SS[.FF] [TZH:TZM]` or `YYYY-MM-DD` or `HH24:MI:SS[.FF][TZH:TZM]` format. `TZH` and `TZM` stands for time zone hour and minute, and `FF` stands for microseconds. If you specify `[TZH:TZM]`, then the `crsctl` command assumes UTC as time zone. If you do not specify `[TZH:TZM]`, then the `crsctl` command assumes the local time zone of the cluster node from where the `crsctl` command is run. Use this parameter with `-aftertime` to query the activities logged at a specific time interval.
`-duration "time_interval" | -follow`	Use `-duration` to specify a time interval that you want to query when you use the `-aftertime` parameter. Specify the timestamp in the `DD HH:MM:SS` format. Use `-follow` to display a continuous stream of activities as they occur.
`-filter "filter_expression "`	Query any number of fields in the cluster resource activity log using the `-filter` parameter. To specify multiple filters, use a comma-delimited list of filter expressions surrounded by double quotation marks ("").
`-fullfmt | -xmlfmt`	To display cluster resource activity log data, choose full or XML format.

Cluster Resource Activity Log Fields

Query any number of fields in the cluster resource activity log using the `-filter` parameter.

Table D-2 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, CRSCTL 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.

Table D-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
`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.
`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 `crsctl stop crs` on a node, then the stack stops and resources bounce. Running the `crsctl stop crs` command generates activities and logged in the `calog`. Since this is a planned action, the `is_planned` field is set to true (1). Otherwise, the `is_planned` 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.

Table D-2 (Cont.) Cluster Resource Activity Log Fields

Field	Description	Use Case
`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 `ora.***`, is started or stopped, for example, then all those activities are logged in the cluster resource activity log. Since `ora.***` is an Oracle entity, the `entity_isoraentity` field is set to true (1). Otherwise the `entity_isoraentity` 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 *server*, 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, *foo* for which the cluster activity log is written.	Query all the cluster activities that match a specific entity name.
`entity_hostname`	The name of the host, for example, `node1`, 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, *cluster1* 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-mb1
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_clustername      : cluster1-mb1
----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_clustername>cluster1-mb1</writer_clustername>
    <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_clustername>cluster1-mb1</entity_clustername>
  </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_clustername>cluster1-mb1</writer_clustername>
    <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>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_clustername>cluster1-mb1</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(node1)=StartCompleted~'" -fullfmt
```

or

```
$ crsctl query calog -filter "(writer_user==root) AND (customer_data==
  'GEN_RESTART@SERVERNAME(node1)=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         : node1
writer_clustername      : cluster1-mb1
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     : node1
entity_isoraentity      : 1
entity_type             : resource
entity_name             : ora.testdb.db
entity_hostname         : node1
entity_clustername      : cluster1-mb1
----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 :
```

D.2 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.
```

D.3 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.
```

D.4 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
```

D.5 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
```

E

chactl Command Reference

The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

E.1 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 E-1 chactl monitor Command Parameters

Parameter	Description
`db_unique_name`	Specify the name of the database.
`model_name`	Specify the name of the model.
`force`	Use the `-force` option to monitor with the specified model without stopping monitoring the target. Without the `-force` option, run `chactl unmonitor` first, and then `chactl monitor` 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.

E.2 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
```

E.3 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.

E.4 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
```

E.5 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
- `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

Syntax

```
chactl calibrate {cluster|database -db db_unique_name} -model model_name
[-force] [-timeranges 'start=time_stamp,end=time_stamp,...']
[-kpiset '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'
-kpiset 'name=CPUPERCENT min=10 max=60'
```

Error Messages

Error: `input string “xc#? %” is misconstructed`

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 misconstructed

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.

E.6 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
```

```
insufficent 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.

E.7 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
...
```

E.8 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
```

E.9 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,...'] [-kpiset '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%
```

E.10 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.

E.11 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.

E.12 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
```

E.13 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 E-1 Example

```
$ chactl import model -name weekday -file /tmp//weekday.mod
```

E.14 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.

E.15 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
```

F

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`, run the `tfactl access` command.
- To use `tfactl` as a command-line tool, run `tfactl` *`command`* `[options]`.
- To use `tfactl` as a shell interface, run `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 (page F-2)
You need root access to `tfactl` or `sudo` access to run all administration commands.

Running Summary and Analysis Commands (page F-7)
Use the summary and analysis commands to view the summary of Oracle Trace File Analyzer deployment, changes detected, events detected by Oracle Trace File Analyzer, and the tool status.

Running Diagnostic Collection Commands (page F-17)
Run the diagnostic collection commands to collect diagnostic data.

F.1 Running Administration Commands

You need root access to `tfactl` or `sudo` access to run all administration commands.

Basic TFACTL commands include:

- **`tfactl start`**: Starts the Oracle Trace File Analyzer daemon on the local node.
- **`tfactl stop`**: Stops the Oracle Trace File Analyzer daemon on the local node.
- **`tfactl enable`**: Enables automatic restart of the Oracle Trace File Analyzer daemon after a failure or system reboot.
- **`tfactl disable`**: Stops any running Oracle Trace File Analyzer daemon and disables automatic restart.
- **`tfactl uninstall`**: Removes Oracle Trace File Analyzer from the local node.
- **`tfactl syncnodes`**: Generates and copies Oracle Trace File Analyzer certificates from one Oracle Trace File Analyzer node to other nodes.
- **`tfactl restrictprotocol`**: Restricts the use of certain protocols.
- **`tfactl status`**: Checks the status of an Oracle Trace File Analyzer process. The output is same as `tfactl print status`.

tfactl diagnosetfa (page F-2)
Use the `tfactl diagnosetfa` command to collect Oracle Trace File Analyzer diagnostic data from the local node to help diagnose issues with Oracle Trace File Analyzer.

tfactl host (page F-3)
Use the `tfactl host` command to add hosts to, or remove hosts from the Oracle Trace File Analyzer configuration.

tfactl set (page F-3)
Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

tfactl access (page F-5)
Use the `tfactl access` command to allow non-root users to have controlled access to Oracle Trace File Analyzer and to run diagnostic collections.

F.1.1 tfactl diagnosetfa

Use the `tfactl diagnosetfa` command to collect Oracle Trace File Analyzer diagnostic data from the local node to help diagnose issues with Oracle Trace File Analyzer.

Syntax

```
tfactl diagnosetfa [-repo repository] [-tag tag_name] [-local]
```

Parameters

Table F-1 tfactl diagnosetfa Command Parameters

Parameter	Description
`-repo repository`	Specify the repository directory for Oracle Trace File Analyzer diagnostic collections.
`-tag tag_name`	Oracle Trace File Analyzer collects the files into `tag_name` directory.
`-local`	Runs Oracle Trace File Analyzer diagnostics only on the local node.

F.1.2 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.

F.1.3 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] [reposizeMB=number] [repositorydir=directory]
[logsize=n [-local]] [logcount=n [-local]] [-c]
```

Parameters

Table F-2 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 `-c` 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).
`trimfiles=ON | OFF`	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 `tfactl diagcollect`, you determine the time range for trimming with the parameters you specify. Oracle recommends that you *not* set this parameter to `OFF`, because untrimmed data can consume much space.
`tracelevel=COLLECT | SCAN | INVENTORY | OTHER: 1 | 2 | 3 | 4`	You can set trace levels for certain operations, including `INVENTORY:n`, `SCAN:n`, `COLLECT:n`, `OTHER:n`. In this syntax, *n* 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.
`diskUsageMon=ON| OFF`	Turns ON (default) or OFF monitoring disk usage and recording snapshots. Oracle Trace File Analyzer stores the snapshots under `tfa/repository/suptools/node/managelogs/usage_snapshot/`.
`diskUsageMonInterval=minutes`	Specify the time interval between snapshots (60 minutes by default).
`manageLogsAutoPurge=ON | OFF`	Turns automatic purging on or off (ON by default in DSC and OFF by default elsewhere).

Table F-2 (Cont.) tfactl set Command Parameters

Parameter	Description
`manageLogsAutoPurgePolicyAge=`*`n`*`d|h`	Age of logs to be purged (30 days by default).
`manageLogsAutoPurgeInterval=`*`minutes`*	Specify the purge frequency (default is 60 minutes).
`reposizeMB=`*`number`*	Sets the maximum size, in MB, of the collection repository.
`repositorydir=`*`directory`*	Specify the collection repository directory.
`logsize=`*`n`* `[-local]`	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 `-local` parameter to apply the change only to the local node.
`logcount=`*`n`* `[-local]`	Sets the maximum number of logs of specified size that Oracle Trace File Analyzer retains (default is 10). Use the `-local` parameter to apply the change only to the local node.
`-c`	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 reposizeMB=20480
```

F.1.4 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.

In Oracle Trace File Analyzer, 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 F-3 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 `–local` 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 that were granted access to Oracle Trace File Analyzer is stored, if the access to non-root users is enabled later. Use the `–local` 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.
`unblock`	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 `tfactl access block`.
`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

The following command adds a user, for example, *abc* to the Oracle Trace File Analyzer access list and enables access to Oracle Trace File Analyzer across cluster.

```
/u01/app/tfa/bin/tfactl access add -user abc
```

The following command adds all members of a group, for example, *xyz* to the Oracle Trace File Analyzer access list and enables access to Oracle Trace File Analyzer on the localhost.

```
/u01/app/tfa/bin/tfactl access add -group xyz -local
```

The following command removes a user, for example, *abc* from the Oracle Trace File Analyzer access list.

```
/u01/app/tfa/bin/tfactl access remove -user abc
```

The following command blocks a user, for example, *xyz* from accessing Oracle Trace File Analyzer.

```
/u01/app/tfa/bin/tfactl access block -user xyz
```

The following command removes all Oracle Trace File Analyzer users and groups.

```
/u01/app/tfa/bin/tfactl access removeall
```

F.2 Running Summary and Analysis Commands

Use the summary and analysis commands to view the summary of Oracle Trace File Analyzer deployment, changes detected, events detected by Oracle Trace File Analyzer, and the tool status.

F.2.1 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 |         |
'-----------------------------------------------+------+------------
+------------------+-----------+---------'
```

F.2.2 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%
```

F.2.3 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.
Jul/25/2016 06:24:43 :
           [db.apxcmupg.apxcmupg_1] : Instance terminated by USER, pid = 21581
Jul/25/2016 06:24:43 :
           [db.rdb11204.rdb112041] : Instance terminated by USER, pid = 18683
Jul/25/2016 06:24:44 :
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "FRA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "FRA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "FRA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "FRA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "FRA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "DATA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "DATA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "DATA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "DATA" does not exist or is not mounted
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15001: diskgroup "DATA" does not exist or is not mounted
Jul/25/2016 06:24:53 :
           [db.+ASM1] : ORA-15032: not all alterations performed
           [db.+ASM1] : ORA-15027: active use of diskgroup "VDATA" precludes its
```

```
dismount
Jul/25/2016 06:25:22 :
          [db.+ASM1] : Shutting down instance (immediate)
          [db.+ASM1] : Shutting down instance: further logons disabled

Summary :
=========
INFO   : 2
ERROR   : 26
WARNING   : 1
```

F.2.4 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.

Filter the output of the command by component, error type, and time.

With the `tfactl analyze` command, you can choose from the following types of log file analysis:

- **Show the most common messages within the logs**: This analysis provides a quick indication of where larger issues are occurring. Oracle Trace File Analyzer takes important messages out of the alert logs and strips the extraneous information from the log messages, organizes the most commonly occurring messages, and displays them in the order from most common to least common. By default, Oracle Trace File Analyzer analyzes error messages, but you can specify a particular type of message for analysis.
- **Search for text within log messages**: This is similar to using the `grep` utility to search, only faster because Oracle Trace File Analyzer checks the time of each 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] [-since 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 F-4 tfactl analyze Command Parameters

Parameter	Description
`-search "pattern"`	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. 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. 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.
`-comp db | asm | crs | acfs | os | osw | oswslabinfo | all`	Select which components you want Oracle Trace File Analyzer to analyze. Default is `all`. • `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 • `oswlabinfo`: OSW Slabinfo output When `OSWatcher` data is available, `OSW` and `OSWSLABINFO` components provide summary views of OSWatcher data.
`-type error | warning | generic`	Select what type of messages Oracle Trace File Analyzer analyzes. Default is `error`.
`-since n[h|d]`	Specify an amount of time, in hours or days, before current time that you want Oracle Trace File Analyzer to analyze.
`-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 F-5 tfactl analyze -type Parameter Arguments

Argument	Description

`error`	Error message patterns for database and Oracle ASM alert logs:
	`.*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`
	Error message patterns for Oracle Grid Infrastructure alert logs:
	`.*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:.*`

`warning`	Warning message patterns for database and Oracle ASM alert logs:
	`NOTE: process .* initiating offline of disk .*`
	`.*WARNING: cache read a corrupted block group.*`
	`.*NOTE: a corrupted block from group FRA was dumped to`

`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" -since 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 -since 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 -since 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 -since 1d`

  Oracle Trace File Analyzer displays a summary of events from system messages from the past day.

- `$ tfactl analyze -since 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 -since 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.

F.2.5 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 F-6 tfactl run Command Parameters

Parameter	Description
`inventory`	Inventory of all trace file directories.
`scan`	Runs a one off scan.
`tool`	Runs the desired analysis tool.

Analysis Tools

Table F-7 tfactl run Analysis Tools Parameters

Parameter	Description
`changes`	Prints system changes.

Table F-7 (Cont.) tfactl run Analysis Tools Parameters

Parameter	Description
`events`	Lists all important events in system.
`exachk`	Runs `Oracle EXAchk`.
`grep`	`grep` for input string in logs.
`history`	Lists commands run in current Oracle Trace File Analyzer shell session.
`ls`	Searches files in Oracle Trace File Analyzer.
`orachk`	Runs `Oracle ORAchk`.
`oratop`	Runs `oratop`.
`oswbb`	Runs `OSWatcher Analyzer`.
`param`	Prints parameter value.
`ps`	Finds a process.
`pstack`	Runs `pstack` on a process.
`prw`	Runs `Procwatcher`.
`sqlt`	Runs `SQLT`.
`summary`	Prints system summary.
`tail`	Tails log files.
`vi`	Searches and opens files in the `vi` editor.

Profiling Tools

Table F-8 tfactl run Profiling Tools Parameters

Parameter	Description
`dbglevel`	Sets CRS log and trace levels using profiles.

F.2.6 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 where.

Table F-9 tfactl toolstatus Output

Host	Tool	Status
hostname	`alertsummary`	DEPLOYED
hostname	`exachk`	DEPLOYED
hostname	`ls`	DEPLOYED
hostname	`triage`	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

F.3 Running Diagnostic Collection Commands

Run the diagnostic collection commands to collect diagnostic data.

F.3.1 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:

https://support.oracle.com/rs?type=doc&id=1513912.2

Syntax

```
tfactl diagcollect [-all | [component_name1] [component_name2] ...
[component_nameN]]  [-node all|local|n1,n2,..] [-tag description] [-z filename] [-
```

```
since 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 *description*	Use this parameter to create a subdirectory for the resulting collection in the Oracle Trace File Analyzer repository.
`-z file_name`	Use this parameter to specify an output file name.
`-since numberh | d | -from "mmm/dd/yyyy hh:mm:ss" -to "mmm/dd/yyyy hh:mm:ss" | -for "mmm/dd/yyyy hh:mm:ss"`	• Specify the `-since` parameter to collect files that have relevant data for the past specific number of hours (*h*) or days (*d*). By default, using the command with this parameter also trims files that are large and shows files only from the specified interval. • Specify the `-from` and `-to` 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 `-for` 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 `-for` is included. No data trimming is done for this option. **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.
`-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

Option	Description
`-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` *`dir1,dir2,...dirn`*	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 `diagcollect` usage examples.

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 –since 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 -since 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 -since 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"
  ```

F.3.2 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 F-10 tfactl directory Command Parameters

Parameter	Description
`-public`	Use the `-public` parameter to make the files contained in the directory available for collection by any Oracle Trace File Analyzer user.
`-private`	Use the `-private` 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 `-exclusions` 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 `-noexclusions` parameter to specify that files in this directory are eligible for collection if the files satisfy time range restrictions.
`-collectall`	Use the `-collectall` parameter to specify that files in this directory are eligible for collection irrespective of type and time range when the user specifies the `-collectalldirs` parameter with the `tfactl diagcollect` 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/+ASM1/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/+ASM1/trace -private
```

The following command removes a directory from all nodes in the cluster:

```
# tfactl directory remove /u01/app/grid/diag/asm/+ASM1/trace -node all
```

F.3.3 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 F-11 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.

Table F-11 (Cont.) tfactl ips Command Parameters

Parameter	Description
`CREATE PACKAGE`	Creates a package, and optionally select contents for the package.
`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 (`zip` 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 (page F-26)
Use the `tfactl ips ADD` command to add incidents to an existing package.

tfactl ips ADD FILE (page F-27)
Use the `tfactl ADD FILE` command to add a file to an existing package.

tfactl ips SHOW CONFIGURATION (page F-34)
Use the `tfactl ips SHOW CONFIGURATION` command to view the current Incident Packaging Service settings.

tfactl ips SHOW PACKAGE (page F-34)
Use the `tfactl ips SHOW PACKAGE` command to view the details of a specific package.

tfactl ips SHOW FILES PACKAGE (page F-35)
Use the `tfactl ips SHOW FILES PACKAGE` command to view the files included in a specific package.

tfactl ips SHOW INCIDENTS PACKAGE (page F-35)
Use the `tfactl ips SHOW INCIDENTS PACKAGE` command to view the incidents included in a specific package.

tfactl ips SHOW PROBLEMS (page F-35)
Use the `tfactl ips SHOW PROBLEMS` command to view the problems for the current Automatic Diagnostic Repository home.

tfactl ips UNPACK FILE (page F-35)
Use the `tfactl ips UNPACK FILE` command to unpack a physical file into a specific path.

tfactl ips UNPACK PACKAGE (page F-36)
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.

F.3.3.1 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 F-12 tfactl ips ADD Command Parameters

Parameter	Description
incid	Specify the ID of the incident to add to the package contents.
prob_id	Specify the ID of the problem to add to the package contents.
prob_key	Specify the problem key to add to the package contents.
seconds	Specify the number of seconds before now for adding package contents.
start_time	Specify the start of time range to look for incidents in.
end_time	Specify the end of time range to look for incidents in.

F.3.3.2 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 F-13 tfactl ips ADD FILE Command Parameters

Parameter	Description
file_spec	Specify the file with file and path (full or relative).
package_id	Specify the ID of the package to add the file to.

F.3.3.3 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 F-14 tfactl ips COPY IN FILE Command Parameters

Parameter	Description
file	Specify the file with file name and full path (full or relative).
new_name	Specify a name for the copy of the file.
pkgid	Specify the ID of the package to associate the file with.
incid	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.

F.3.3.4 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 F-15 tfactl ips REMOVE Command Parameters

Parameter	Description
incid	Specify the ID of the incident to add to the package contents.
prob_id	Specify the ID of the problem to add to the package contents.
prob_key	Specify the problem key to add to the package contents.

Example

```
$ tfactl ips remove incident 22 package 12
```

F.3.3.5 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
```

F.3.3.6 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 F-16 tfactl ips ADD NEW INCIDENTS PACKAGE Command Parameters

Parameter	Description
package_id	Specify the ID of the package to add the incidents to.

F.3.3.7 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 F-17 tfactl ips GET REMOTE KEYS FILE Command Parameters

Parameter	Description
file_spec	Specify the file with file name and full path (full or relative).
package_id	Specify the ID of the package to get keys for.

Example

```
$ tfactl ips get remote keys file /tmp/key_file.txt package 12
```

F.3.3.8 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
```

F.3.3.9 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 F-18 tfactl ips CREATE PACKAGE Command Parameters

Parameter	Description
incid	Specify the ID of the incident to use for selecting the package contents.
prob_id	Specify the ID of the problem to use for selecting the package contents.
prob_key	Specify the problem key to use for selecting the package contents.

Table F-18 (Cont.) tfactl ips CREATE PACKAGE Command Parameters

Parameter	Description
seconds	Specify the number of seconds before now for selecting the package contents.
start_time	Specify the start of time range to look for the incidents in.
end_time	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'
```

F.3.3.10 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
```

F.3.3.11 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 F-19 tfactl ips GENERATE PACKAGE Command Parameters

Parameter	Description
package_id	Specify the ID of the package to create physical package file for.
path	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
```

F.3.3.12 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 F-20 tfactl ips DELETE PACKAGE Command Parameters

Parameter	Description
package_id	Specify the ID of the package to delete.

Example

```
$ tfactl ips delete package 12
```

F.3.3.13 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 F-21 tfactl ips GET MANIFEST FROM FILE Command Parameters

Parameter	Description
file	Specify the external file with file name and full path.

Example

```
$ tfactl ips GET MANIFEST FROM FILE /tmp/IPSPKG_200704130121_COM_1.zip
```

F.3.3.14 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
```

F.3.3.15 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 F-22 tfactl ips PACK Command Parameters

Parameter	Description
incid	Specify the ID of the incident to use for selecting the package contents.
prob_id	Specify the ID of the problem to use for selecting the package contents.
prob_key	Specify the problem key to use for selecting the package contents.
seconds	Specify the number of seconds before the current time for selecting the package contents.
start_time	Specify the start of time range to look for the incidents in.
end_time	Specify the end of time range to look for the incidents in.
path	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'
```

F.3.3.16 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 F-23 tfactl ips SET CONFIGURATION Command Parameters

Parameter	Description
parameter_id	Specify the ID of the parameter to change.
value	Specify the new value for the parameter.

Example

```
$ tfactl ips set configuration 6 2
```

F.3.3.17 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
```

F.3.3.18 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
```

F.3.3.19 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
```

F.3.3.20 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
```

F.3.3.21 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
```

F.3.3.22 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
```

F.3.3.23 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
```

F.3.4 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.

F.3.5 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 F-24 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.

Table F-24 (Cont.) tfactl print Command Parameters

Parameter	Description
`actions`	Lists all the actions submitted to Oracle Trace File Analyzer, such as diagnostic collection. By default, `tfactl print` 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 `tfactl purge` command to clear collections from the repository.
`cookie`	Generates and displays an identification code for use by the `tfactl set` 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.

F.3.6 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
```

F.3.7 tfactl managelogs

Use the `tfactl managelogs` command to manage Automatic Diagnostic Repository log and trace files.

Syntax

```
tfactl managelogs [-purge [[-older nm|h|d] | [-gi] | [-database all|d1,d2,...]]]
[-show [usage|variation] [[-older nd] | [-gi] | [-database all|d1,d2,...]]]
```

Parameters

Table F-25 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 `purge` command.

Table F-26 tfactl managelogs Show Options

Show Option	Description
`-older`	Time period for change in log volume.

Table F-26 (Cont.) tfactl managelogs Show Options

Show Option	Description
`-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/tnslsnr/node1/listener_scan2/trace               | 12.00
KB  | 12.00 KB  |
+-----------------------------------------------------------------------------
+-----------+-----------+
```

```
| /scratch/app/orabase/diag/tnslsnr/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/tnslsnr/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/tnslsnr/node1/listener_scan2/alert               | 16.00
KB  | 16.00 KB  |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/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/tnslsnr/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/tnslsnr/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/tnslsnr/node1/listener_scan3/alert               |
562.34 MB | 726.93 MB |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/node1/listener_scan1/incident            | 4.00
KB   | 4.00 KB   |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/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/tnslsnr/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/tnslsnr/node1/listener_scan1/trace                  | 8.00
KB   | 8.00 KB   |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/node1/listener_scan1/cdump                  | 4.00
KB   | 4.00 KB   |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/node1/listener_scan3/trace                  |
263.64 MB | 340.29 MB |
+-------------------------------------------------------------------------------
+-----------+-----------+
| /scratch/app/orabase/diag/tnslsnr/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/tnslsnr/node1/listener_scan2/cdump               | 4.00
KB   | 4.00 KB   |
'-------------------------------------------------------------------------------
+-----------+-----------'
```

Index

A

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V