Autonomous Health Framework

User’s Guide

21c
F31833-05
February 2025

Autonomous Health Framework User’s Guide, 21c

F31833-05

Primary Authors: Nirmal Kumar, Janet Stern

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

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

Contents

Preface

1 Introduction to Oracle Autonomous Health Framework

Part I Analyzing the Cluster Configuration

2 Analyzing Risks and Complying with Best Practices

3 Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

Part II Automatically Monitoring the Cluster

4 Collecting Operating System Resources Metrics

5 Monitoring System Metrics for Cluster Nodes

Part III Automatic Problem Solving

6 Resolving Database and Database Instance Delays

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

7 Using Automatic Diagnostic Collections

8 Using On-Demand Diagnostic Collections

9 Using REST Service

10 Managing and Configuring Oracle Trace File Analyzer

11 Managing Oracle Database and Oracle Grid Infrastructure Logs

Part V Appendixes

A Compliance Framework (Oracle ORAchk and Oracle EXAchk) Command-Line Options

B OCLUMON Command Reference

C Managing the Cluster Resource Activity Log

Preface

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

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

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

This Preface contains these topics:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

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

Documentation Accessibility

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

Access to Oracle Support

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

Related Documentation

For more information, see the following Oracle resources:

Related Topics

- *Oracle Automatic Storage Management Administrator's Guide*
- *Oracle Database 2 Day DBA*

- *Oracle Database Concepts*
- *Oracle Database Examples Installation Guide*
- *Oracle Database Licensing Information User Manual*
- *Oracle Database Release Notes*
- *Oracle Database Upgrade Guide*
- *Oracle Grid Infrastructure Installation and Upgrade Guide*
- *Oracle Real Application Clusters Installation Guide for Linux and UNIX*
- *Oracle Real Application Clusters Installation Guide for Microsoft Windows x64 (64-Bit)*

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.

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.

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

1.1 Oracle Autonomous Health Framework Problem and Solution Space

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

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

- Availability Issues
  Availability issues are runtime issues that threaten the availability of software stack.
- Performance Issues
  Performance issues are runtime issues that threaten the performance of the system.

1.1.1 Availability Issues

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

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

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

Examples of Server Availability Issues

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

Examples of such issues are:

- **Issue:** Memory stress caused by a server running out of free physical memory, results in the operating system `Swapper` process to run for extended periods of time moving memory to disk. Swapping prevents time-critical cluster processes from running and eventually causing the node to be evicted.

  Solution: Memory Guard detects the memory stress in advance and causes work to be drained to free up memory.

- **Issue:** Network congestion on the private interconnect can cause time-critical internode or storage I/O to have excessive latency or dropped packets. This type of failure typically builds up and can be detected early, and corrected or relieved.

  Solution: If a change in the server configuration causes this issue, then Cluster Verification Utility (CVU) detects it if the issue persists for more than an hour. However, Oracle Cluster Health Advisor detects the issue within minutes and presents corrective actions.

- **Issue:** Network failures on the private interconnect caused by a pulled cable or failed network interface card (NIC) can immediately result in evicted nodes.

  Solution: Although these types of network failures cannot be detected early, the cause can be narrowed down by using Cluster Health Monitor and Oracle Trace File Analyzer to pinpoint the time of the failure and the network interfaces involved.

Examples of Database Availability Issues

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

Examples of such issues are:

- **Issue:** Runaway queries or hangs can deny critical database resources such as locks, latches, or CPU to other sessions. Denial of critical database resources results in database or an instance of a database being non-responsive to applications.

  Solution: Hang Manager detects and automatically resolves these types of hangs. Also, Oracle Cluster Health Advisor detects, identifies, and notifies the database administrator of such hangs and provides an appropriate corrective action.

- **Issue:** Denial-of-service (DoS) attacks, vulnerabilities, or simply software bugs can cause a database or a database instance to be unresponsive.

  Solution: Proactive recommendations of known issues and their resolutions provided by Oracle ORAchk can prevent such occurrences. If these issues are not prevented, then automatic collection of logs by Oracle Trace File Analyzer, in addition to data collected by Cluster Health Monitor, can speed up the correction of these issues.

- **Issue:** Configuration changes can cause database outages that are difficult to troubleshoot. For example, incorrect permissions on the `oracle.bin` file can prevent session processes from being created.

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

1.1.2 Performance Issues

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

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

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

Examples of Database Server Performance Issues

- **Issue:** Deviations from best practices in configuration can cause database server performance issues.

  Solution: Oracle ORAchk detects configuration issues when Oracle ORAchk runs periodically and notifies the database administrator of the appropriate corrective settings.
- **Issue:** A session can cause other sessions to slow down waiting for the blocking session to release its resource or complete its work.

  Solution: Blocker Resolver detects these chains of sessions and automatically terminates 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:** 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.

1.2 Components of Autonomous Health Framework

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

- Introduction to Oracle Autonomous Health Framework Configuration Audit Tools
  Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.
- Introduction to Cluster Health Monitor
  Cluster Health Monitor is a component of Oracle Grid Infrastructure, which continuously monitors and stores Oracle Clusterware and operating system resources metrics.
- Introduction to Oracle Trace File Analyzer
  Oracle Trace File Analyzer is a utility for targeted diagnostic collection that simplifies diagnostic data collection for Oracle Clusterware, Oracle Grid Infrastructure, and Oracle

Real Application Clusters (Oracle RAC) systems, in addition to single instance, non-clustered databases.

- Introduction to Oracle Cluster Health Advisor
  Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.
- Introduction to Memory Guard
  Memory Guard is an Oracle Real Application Clusters (Oracle RAC) environment feature to monitor the cluster nodes to prevent node stress caused by the lack of memory.
- Introduction to Hang Manager
  Hang Manager is an Oracle Real Application Clusters (Oracle RAC) environment feature that autonomously resolves delays and keeps the resources available.

1.2.1 Introduction to Oracle Autonomous Health Framework Configuration Audit Tools

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

1.2.2 Introduction to Cluster Health Monitor

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

Enabled by default, Cluster Health Monitor:

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

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

Related Topics

- Collecting Operating System Resources Metrics
  CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.

1.2.3 Introduction to Oracle Trace File Analyzer

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

Enabled by default, Oracle Trace File Analyzer:

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

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

New Attention Log for Efficient Critical Issue Resolution

Diagnosability of database issues is enhanced through a new attention log, as well as classification of information written to database trace files. The new attention log is written in a structured format (XML or JSON) that is much easier to process or interpret and only contains information that requires attention from an administrator. The contents of trace files now contains information that enables much easier classification of trace messages, such as for security and sensitivity.

Enhanced diagnosability features simplify database administration and improve data security.

For more information, see Attention Log

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.

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.

1.2.6 Introduction to Hang Manager

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

Enabled by default, Hang Manager:

- Reliably detects database delays and deadlocks
- Autonomously resolves database delays and deadlocks
- Logs all detections and resolutions
- Provides SQL interface to configure sensitivity (Normal/High) and trace file sizes

A database delays 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 delays.

Related Topics

- Resolving Database and Database Instance Delays
  Hang Manager preserves the database performance by resolving delays and keeping the resources available.

Part I

Analyzing the Cluster Configuration

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

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

2 Analyzing Risks and Complying with Best Practices

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

- Using Oracle ORAchk and Oracle EXAchk to Automatically Check for Risks and System Health
  Oracle recommends that you use the daemon process to schedule recurring health checks at regular intervals.
- Email Notification and Health Check Report Overview
  The following sections provide a brief overview about email notifications and sections of the HTML report output.
- Configuring Oracle ORAchk and Oracle EXAchk
  To configure Oracle ORAchk and Oracle EXAchk, use the procedures explained in this section.
- Using Oracle ORAchk and Oracle EXAchk to Manually Generate Compliance Check Reports
  This section explains the procedures to manually generate compliance check reports.
- Managing the Oracle ORAchk and Oracle EXAchk Daemons
  This section explains the procedures to manage Oracle ORAchk and Oracle EXAchk daemons.
- Tracking Support Incidents
  The **Incidents** tab gives you a complete system for tracking support incidents.
- Tracking File Attribute Changes and Comparing Snapshots
  Use the Oracle ORAchk and Oracle EXAchk `-fileattr` option and command flags to record and track file attribute settings, and compare snapshots.
- Collecting and Consuming Health Check Data
  Oracle Health Check Collections Manager for Oracle Application Express 5.x provides you an enterprise-wide view of your health check collection data.
- Integrating Compliance Check Results with Other Tools
  Integrate health check results from Oracle Autonomous Health Framework into Oracle Enterprise Manager and other third-party tools.
- Using Oracle ORAchk to Confirm System Readiness for Implementing Application Continuity
  Application Continuity Checking for Application Continuity enables you to deploy Application Continuity easily and transparently.
- Configuring Oracle REST Data Services (ORDS)
- Using Oracle Autonomous Health Framework Compliance Over REST
  Oracle ORAchk and Oracle EXAchk include full REST support allowing invocation and query over HTTPS.
- Command-Line Options to Generate Password Protected Collection zip Files
  Use the list of commands to encrypt or decrypt diagnostic collection `zip` files.

- Caching Discovery Data
  Use the list of commands to manage caching of discovery data.
- Applying Patch Between Releases
  Use the list of commands to manage patches.
- Creating, Modifying, and Deleting User-Defined Profiles
  Specify a comma-delimited list of check IDs to create and modify custom profiles.
- Sanitizing Sensitive Information in the Diagnostic Collections
  Oracle Autonomous Health Framework uses Adaptive Classification and Redaction (ACR) to sanitize sensitive data.
- Troubleshooting Compliance Framework (Oracle ORAchk and Oracle EXAchk)
  Follow the steps explained in this section to troubleshoot and fix Compliance Framework (Oracle ORAchk / Oracle EXAchk) related issues.

Related Topics

- Introduction to Oracle Autonomous Health Framework Configuration Audit Tools
  Oracle ORAchk and Oracle EXAchk provide a lightweight and non-intrusive health check framework for the Oracle stack of software and hardware components.

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

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

> **Note:**
>
> Daemon mode is supported only on the Linux and Solaris operating systems.

Configure the daemon to:

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

> **Note:**
>
> While running, the daemon answers all the prompts required by subsequent on-demand health checks.
>
> To run on-demand health checks, do not use the daemon process started by others. Run on-demand health checks within the same directory where you have started the daemon.

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

Related Topics

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

2.2 Email Notification and Health Check Report Overview

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

- First Email Notification
  After completing compliance 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 Compliance Check Report Contain?
  Compliance check reports contain the health status of each system grouped under different sections of the report.
- Subsequent Email Notifications
  For the subsequent compliance check runs after the first email notification, the daemon emails the summary of differences between the most recent runs.

Related Topics

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

2.2.1 First Email Notification

After completing compliance 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 Compliance Check Report Contain?

Compliance 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 compliance 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 compliance 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) compliance 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 compliance 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 compliance 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 compliance 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 and Oracle EXAchk
  Run compliance checks as `root`. Also, run compliance checks as the Oracle Database home owner or the Oracle Grid Infrastructure home owner.
- Handling of Root Passwords
  Handling of `root` passwords depends on whether you have installed the Expect utility.
- Configuring Email Notification System
  Oracle Health Check Collections Manager provides an email notification system that users can subscribe to.

2.3.1 Deciding Which User Should Run Oracle ORAchk and Oracle EXAchk

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

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

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

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

Specify a location for sudo remote access as follows:

```
export RAT_ROOT_SH_DIR=/mylocation
```

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

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

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

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

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

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

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

   ```
   export RAT_ROOT_SH_DIR=/mylocation
   ```

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

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

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

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

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

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

   Running compliance 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 compliance checks that need `root` access.

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

   There are several other options:

   - Skip the checks that require `root` access.

- Specify the `root` user ID and password when prompted.
- Configure `sudo`.

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

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

  For example:

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

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

- Pre-configure passwordless SSH connectivity.

2.3.2 Handling of Root Passwords

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

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

To handle root passwords:

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

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

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

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

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

   If you enter the password incorrectly or the password is changed between the time it is entered and used, then Oracle Autonomous Health Framework:

   - Notifies you
   - Skips relevant checks

3. Run the compliance checks after resolving the issues.

   If Oracle Autonomous Health Framework skips any of the compliance checks, then the tools log details about the skipped checks in the report output.

Related Topics

- Expect - Expect - Home Page

2.3.3 Configuring Email Notification System

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

The setup involves:

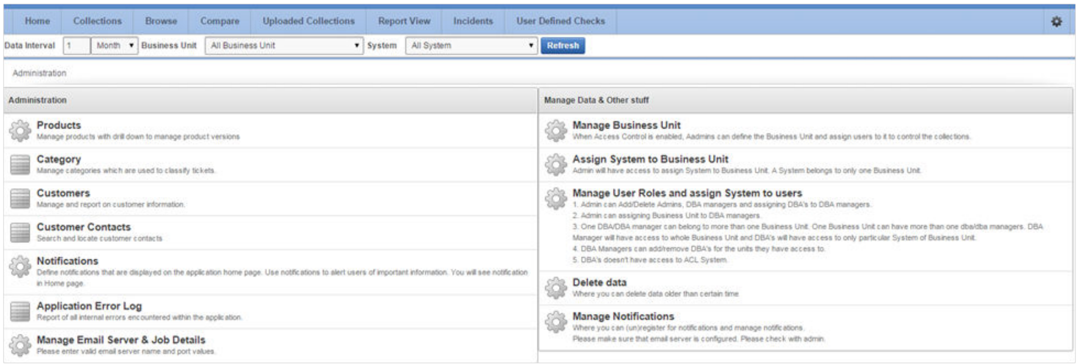
- Configuring the email server, port, and the frequency of email notifications.
- Registering the email address

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

To configure the email notification system:

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

Figure 2-1 Oracle Health Check Collections Manager - Administration



Data Interval 1 Month
Business Unit All Business Unit
System All System

Administration

Products
Manage products with drill down to manage product versions

Category
Manage categories which are used to classify tickets.

Customers
Manage and report on customer information.

Customer Contacts
Search and locate customer contacts

Notifications
Define notifications that are displayed on the application home page. Use notifications to alert users of important information. You will see notification in Home page.

Application Error Log
Report of all internal errors encountered within the application.

Manage Email Server & Job Details
Please enter valid email server name and port values.

Manage Data & Other stuff

Manage Business Unit
When Access Control is enabled, Aadmins can define the Business Unit and assign users to it to control the collections.

Assign System to Business Unit
Admin will have access to assign System to Business Unit. A System belongs to only one Business Unit.

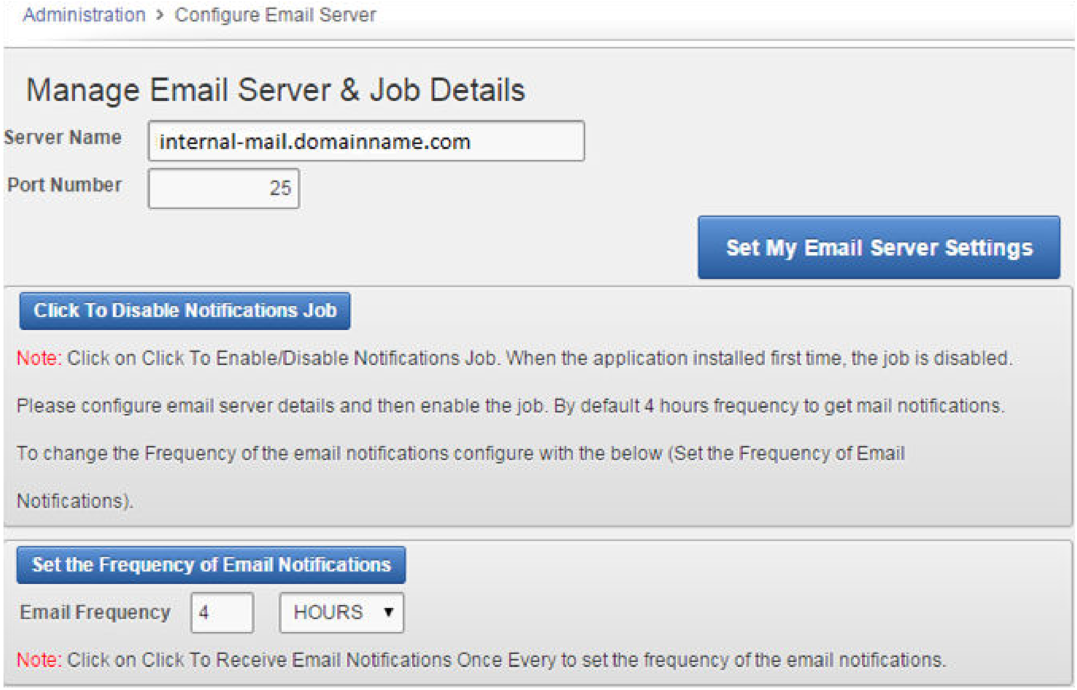
Manage User Roles and assign System to users
1. Admin can Add/Delete Admins, DBA managers and assigning DBA's to DBA managers.
2. Admin can assigning Business Unit to DBA managers.
3. One DBA/DBA manager can belong to more than one Business Unit. One Business Unit can have more than one dba/dba managers. DBA Manager will have access to whole Business Unit and DBA's will have access to only particular System of Business Unit.
4. DBA Managers can add/remove DBA's for the units they have access to.
5. DBA's doesn't have access to ACL System.

Delete data
Where you can delete data older than certain time

Manage Notifications
Where you can (un)register for notifications and manage notifications.
Please make sure that email server is configured. Please check with admin.

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

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



Administration > Configure Email Server

Manage Email Server & Job Details

Server Name: internal-mail.domainname.com

Port Number: 25

Set My Email Server Settings
Click To Disable Notifications Job

Note: Click on Click To Enable/Disable Notifications Job. When the application installed first time, the job is disabled.

Please configure email server details and then enable the job. By default 4 hours frequency to get mail notifications.

To change the Frequency of the email notifications configure with the below (Set the Frequency of Email Notifications).

Email Frequency: 4 HOURS ▼

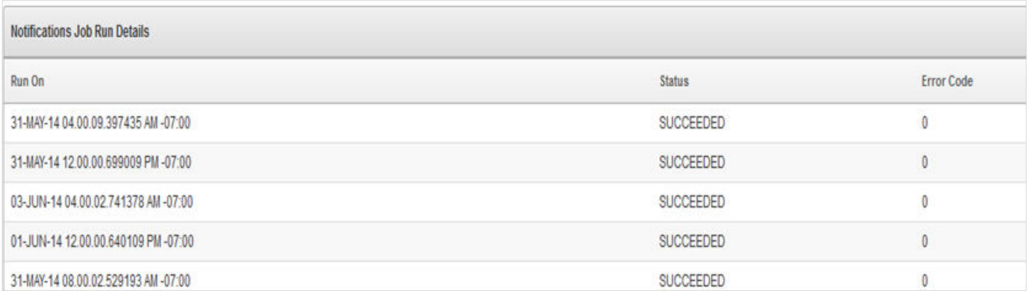
Note: Click on Click To Receive Email Notifications Once Every to set the frequency of the email notifications.

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

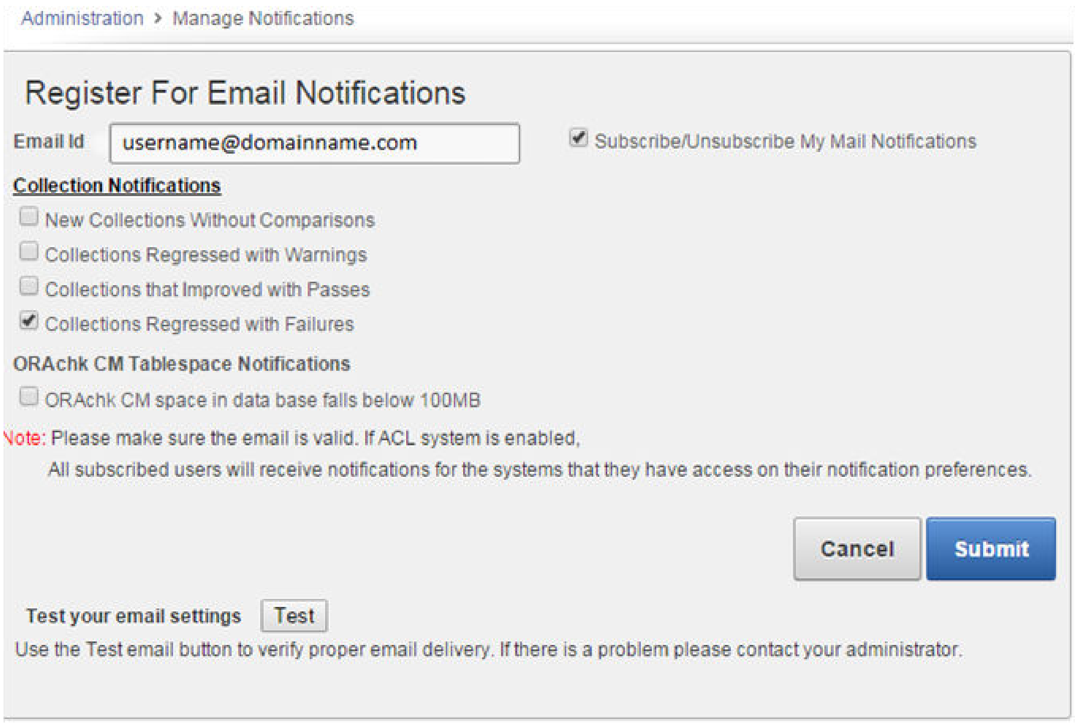


Notifications Job Run Details

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



Register For Email Notifications

Email Id: username@domainname.com

☑ Subscribe/Unsubscribe My Mail Notifications

Collection Notifications

- ☐ New Collections Without Comparisons
- ☐ Collections Regressed with Warnings
- ☐ Collections that Improved with Passes
- ☑ Collections Regressed with Failures

ORAchk CM Tablespace Notifications

- ☐ ORAchk CM space in data base falls below 100MB

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

Test your email settings

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

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:

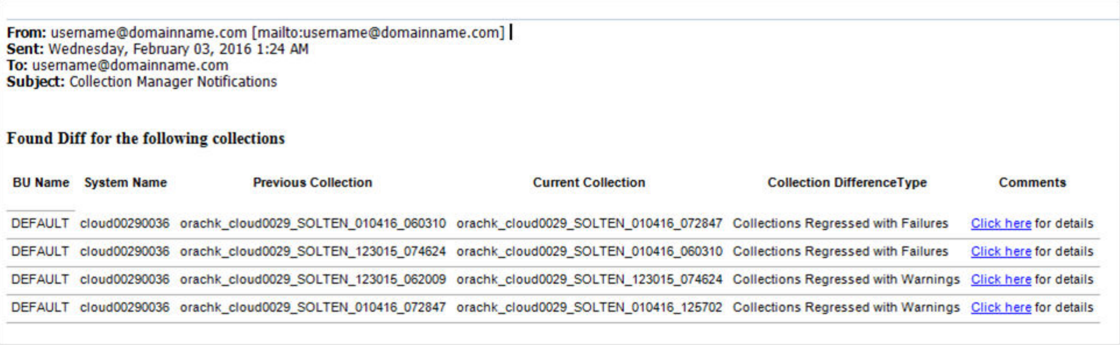
```
From: username@example.com
Sent: Thursday, January 28, 2016 12:21 PM
To: username@example.com
Subject: Test Mail From Collection Manager

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



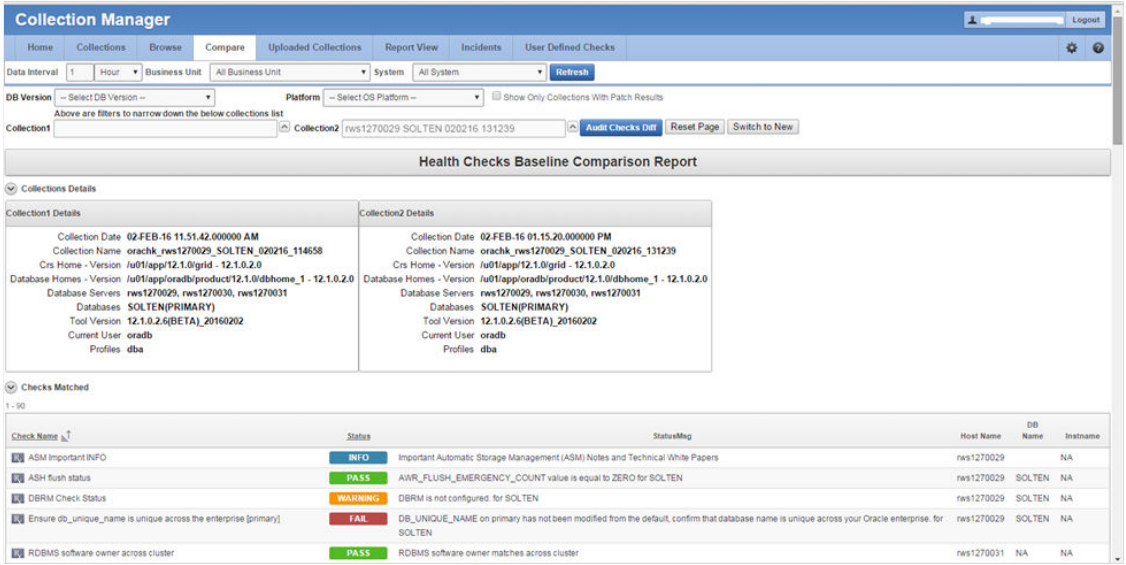
From: username@domainname.com [mailto:username@domainname.com]
Sent: Wednesday, February 03, 2016 1:24 AM
To: username@domainname.com
Subject: Collection Manager Notifications

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



Data Interval 1 Hour | Business Unit All Business Unit | System All System

DB Version -- Select DB Version -- | Platform -- Select OS Platform -- | □ Show Only Collections With Patch Results

Above are filters to narrow down the below collections list

Collection1 | Collection2 rws1270029 SOLTEN 020216 131239

Health Checks Baseline Comparison Report

Collections Details

Collection1 Details		Collection2 Details	
Collection Date	02-FEB-16 11.51.42.000000 AM	Collection Date	02-FEB-16 01.15.20.000000 PM
Collection Name	orachk_rws1270029_SOLTEN_020216_114658	Collection Name	orachk_rws1270029_SOLTEN_020216_131239
Crs Home - Version	/u01/app/12.1.0/grid - 12.1.0.2.0	Crs Home - Version	/u01/app/12.1.0/grid - 12.1.0.2.0
Database Homes - Version	/u01/app/oradb/product/12.1.0/dbhome_1 - 12.1.0.2.0	Database Homes - Version	/u01/app/oradb/product/12.1.0/dbhome_1 - 12.1.0.2.0
Database Servers	rws1270029, rws1270030, rws1270031	Database Servers	rws1270029, rws1270030, rws1270031
Databases	SOLTEN(PRIMARY)	Databases	SOLTEN(PRIMARY)
Tool Version	12.1.0.2.6(BETA)_20160202	Tool Version	12.1.0.2.6(BETA)_20160202
Current User	oradb	Current User	oradb
Profiles	dba	Profiles	dba

Checks Matched

1 - 90

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

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

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

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

2.4.1 Running Compliance Checks On-Demand

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

Examples of when you must run compliance 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 compliance check runs, log in to the system as an appropriate user, and then run an appropriate tool. Specify the options to direct the type of run that you want.

```
$ ./orachk
```

```
$ ./exachk
```

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

Output varies depending on your environment and options used:

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

> **Note:**
>
> If you are prompted for passwords, then the Expect utility runs when available. In this way, the passwords are gathered at the beginning, and the Expect utility supplies the passwords when needed at the root password prompts. The Expect utility being supplying the passwords enables the tool to continue without the need for further input. If you do not use the Expect utility, then closely monitor the run and enter the passwords interactively as prompted.
>
> Without the Expect utility installed, you must enter passwords many times depending on the size of your environment. Therefore, Oracle recommends that you use the Expect utility.

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

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

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

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

After upgrading, run the post-upgrade checks:

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

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

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

Related Topics

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

2.4.2 Running Compliance Checks in Silent Mode

Run compliance 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 compliance checks in silent mode using the `-s` option does not run any checks on the storage servers and switches.

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

When compliance 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 Compliance Checks that Require root Access

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

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

```
$ orachk -s
```

```
$ exachk -s
```

Excluding Compliance Checks that Require root Access

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

```
$ orachk -S
```

```
$ exachk -S
```

Related Topics

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

2.4.3 Running On-Demand With or Without the Daemon

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

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

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

   ```
   $ orachk
   ```

   ```
   $ exachk
   ```

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

   ```
   $ orachk -nodaemon
   ```

   ```
   $ exachk -nodaemon
   ```

> **Note:**
>
> Daemon mode is supported only on the Linux and Solaris operating systems.

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

2.4.4 Generating a Diff Report

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

To identify the changes since the last run:

- Run the following command:

  ```
  $ orachk -diff report_1 report_2
  ```

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

2.4.5 Sending Results by Email

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

To send health check run results by email:

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

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

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

   Where *email_recipients* is a comma-delimited list of email addresses.
2. Verify the email configuration settings using the `-testemail` option.

2.5 Managing the Oracle ORAchk and Oracle EXAchk Daemons

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

- Starting and Stopping the Daemon
  Start and stop the daemon and force the daemon to stop a compliance check run.
- Configuring the Daemon for Automatic Restart
  By default, you must manually restart the daemon if you restart the *server* or *node* on which the daemon is running.
- Setting and Getting Options for the Daemon
  Set the daemon options before you start the daemon. Reset the daemon options anytime after starting the daemon.

- Querying the Status and Next Planned Daemon Run
  Query the status and next automatic run schedule of the running daemon.

2.5.1 Starting and Stopping the Daemon

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

To start and stop the daemon:

1. To start the daemon:

   ```
   $ orachk -d start
   ```

   ```
   $ exachk -d start
   ```

   The tools prompt you to provide required information during startup.

2. To stop the daemon:

   ```
   $ orachk -d stop
   ```

   ```
   $ exachk -d stop
   ```

   If a compliance 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 compliance check run:

   ```
   $ orachk -d stop_client
   ```

   ```
   $ exachk -d stop_client
   ```

The daemon stops the compliance check run and then confirms when it is done. If necessary, then 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.

Configure the daemons to auto restart as `root`.

To configure the daemon to restart automatically:

1. To configure the daemon to restart automatically:

   ```
   $ orachk -initsetup
   ```

   ```
   $ exachk -initsetup
   ```

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

   > **Note:**
   >
   > Stop the daemon before running `-initsetup`, if the daemon is already running.

2. To query automatic restart status of the daemon:

   ```
   $ orachk -initcheck
   ```

   ```
   $ exachk -initcheck
   ```

3. To remove automatic restart configuration:

   ```
   $ 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:

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

  Set an option as follows:

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

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

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

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


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

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

2.5.3.1 AUTORUN_SCHEDULE

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

To schedule recurring compliance check runs:

- Set the `AUTORUN_SCHEDULE` option, as follows:

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

  Where:

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

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

Table 2-1 AUTORUN_SCHEDULE

Example	Result
`"AUTORUN_SCHEDULE=0,15,30,45 * * * *"`	Runs every 15 minutes.
`"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"
```

Optionally, you can specify the name of the profile. If you do not specify, then `id=DEFAULT`.

For example:

```
$ orachk -id dba -set "AUTORUN_SCHEDULE=3 * * 0"


$ exachk -id dba -set "AUTORUN_SCHEDULE=3 * * 0"
```

2.5.3.2 AUTORUN_FLAGS

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

To configure how compliance checks should run:

- 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.

Table 2-2 (Cont.) AUTORUN_FLAGS

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

For example:

```
$ orachk -set "AUTORUN_FLAGS=-profile sysadmin -tag sysadmin"
```

```
$ exachk -set "AUTORUN_FLAGS=-profile sysadmin -tag sysadmin"
```

2.5.3.3 NOTIFICATION_EMAIL

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

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

To configure email notifications:

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"
   ```

   Optionally, you can specify the name of the profile. If you do not specify, then `id=DEFAULT`.

   For example:

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

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

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

   ```
   $ orachk -testemail all
   ```

   ```
   $ exachk -testemail all
   ```

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

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

2.5.3.4 collection_retention

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

To configure collection retention period:

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

   ```
   collection_retention=number_of_days
   ```

   If you do not set this option, then the daemon does not purge the stale collection.
2. Set the `collection_retention` option to an appropriate number of days based on:
   - Frequency of your scheduled collections
   - Size of the collection results
   - Available disk space

For example:

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

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

To Control Collection Retention Using Size

Set the size in MB using the environment variable `RAT_PURGE_SIZE`. When the health check collections consume the size specified, then Oracle ORAchk starts purging the old collections, and retains the space specified using `RAT_PURGE_SIZE`.

For example:

```
$export RAT_PURGE_SIZE=4096
```

2.5.3.5 PASSWORD_CHECK_INTERVAL

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

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

To configure password validation frequency:

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

   ```
   PASSWORD_CHECK_INTERVAL=number_of_hours
   ```

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

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

For example:

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

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

2.5.3.6 Setting Multiple Option Profiles for the Daemon

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

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

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

To set multiple option profiles for the daemon:

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

  Where, *`id`* is the name of the profile

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

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

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

Define the database administrator profile as follows:

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

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

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


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

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

Define the system administrator profile as follows:

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

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


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

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

2.5.3.7 Getting Existing Options for the Daemon

Query the values that you set for the daemon options.

To query the values, use

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

where:

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

To get existing options for the daemon:

1. To get a specific daemon option:

For example:

```
$ ./orachk -get NOTIFICATION_EMAIL

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


$ ./exachk -get NOTIFICATION_EMAIL

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

2. To query multiple daemon option profiles:

   For example:

   ```
   $ ./orachk -get NOTIFICATION_EMAIL

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

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


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


   $ ./exachk -get NOTIFICATION_EMAIL

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

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


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

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

   For example:

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

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

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


$ ./exachk -id dba -get NOTIFICATION_EMAIL

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

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

   For example:

```
$ ./orachk -get all

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


$ ./exachk -get all

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

5. To query all daemon option profiles:

   For example:

```
$ ./orachk -get all

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

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

```
collection_retention = 30
password_check_interval = 1

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

Where:

- `-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:

   ```
   $ 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:

   ```
   $ 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 compliance check run:

   ```
   $ 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 compliance check run of a specific profile using `-id ID -d nextautorun`:

```
$ orachk -d ID -d nextautorun
```

```
$ exachk -d ID -d nextautorun
```

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

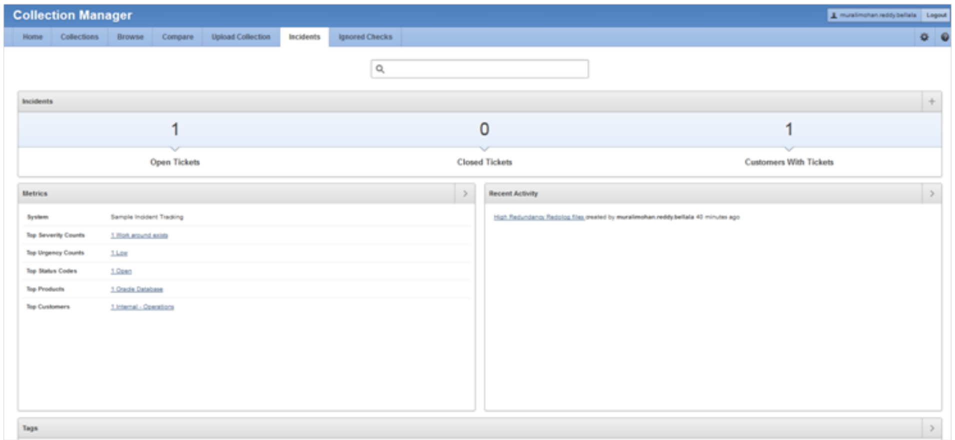
2.6 Tracking Support Incidents

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

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

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

Figure 2-7   Incidents Tab



1 Open Tickets

0 Closed Tickets

1 Customers With Tickets

System	Sample Incident Tracking
Top Severity Counts	1 Work around exists
Top Urgency Counts	1 Low
Top Status Codes	1 Open
Top Products	1 Oracle Database
Top Customers	1 Internal - Operations

High Redundancy Redolog files created by muralimohan.reddy.bellala 40 minutes ago

Incident Tracking Features

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

Related Topics

- Creating or Editing Incidents Tickets
  Create or edit incident tickets for individual checks or for an entire collection.

2.7 Tracking File Attribute Changes and Comparing Snapshots

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

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

- Using the File Attribute Check With the Daemon
  You must have Oracle Grid Infrastructure installed and running before you use `-fileattr`.
- Taking File Attribute Snapshots
  By default, Oracle Grid Infrastructure homes and all the installed Oracle Database homes are included in the snapshots.
- Including Directories to Check
  Include directories in the file attribute changes check.
- Excluding Directories from Checks
  Exclude directories from file attribute changes checks.
- Rechecking Changes
  Compare the new snapshot with the previous one to track changes.

- Designating a Snapshot As a Baseline
  Designate a snapshot as a baseline to compare with other snapshots.
- Restricting System Checks
  Restrict Oracle ORAchk and Oracle EXAchk to perform only file attribute changes checks.
- Removing Snapshots
  Remove the snapshots diligently.

2.7.1 Using the File Attribute Check With the Daemon

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

To use file attribute check with the daemon:

1. Start the daemon.

   ```
   orachk -d start
   ```

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

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


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

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

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

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

   For example:

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

2.7.2 Taking File Attribute Snapshots

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

To take file attribute snapshots:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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:

- 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 5.x provides you an enterprise-wide view of your health check collection data.

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

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

2.8.1 Selectively Capturing Users During Login

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

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

By default, Oracle Health Check Collections Manager:

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

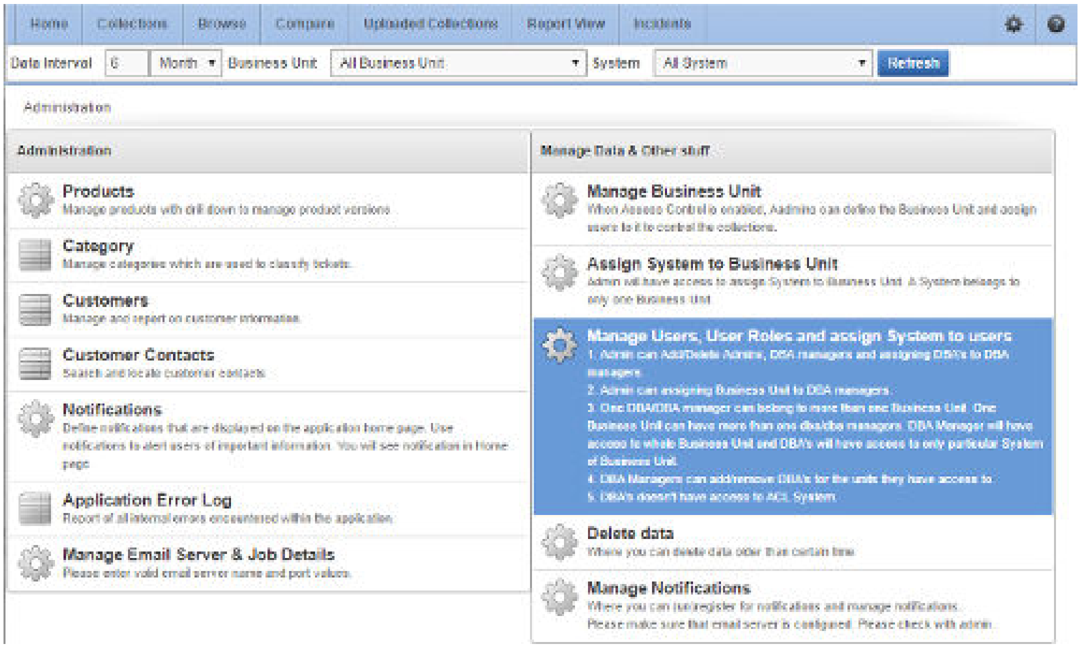
> **Note:**
>
> The Oracle Health Check Collections Manager roles are specific to Oracle Health Check Collections Manager and do not equate to system privileges. For example, the DBA role is not granted SYSDBA system privilege.

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

To enable or disable capturing user details automatically:

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

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



Products
Manage products with drill down to manage product versions

Category
Manage categories which are used to classify tickets.

Customers
Manage and report on customer information.

Customer Contacts
Search and locate customer contacts

Notifications
Define notifications that are displayed on the application home page. Use notifications to alert users of important information. You will see notification in Home page.

Application Error Log
Report of all internal errors encountered within the application.

Manage Email Server & Job Details
Please enter valid email server name and port values.

Manage Business Unit
When Access Control is enabled, Admins can define the Business Unit and assign users to it to control the collections.

Assign System to Business Unit
Admin will have access to assign System to Business Unit. A System belongs to only one Business Unit

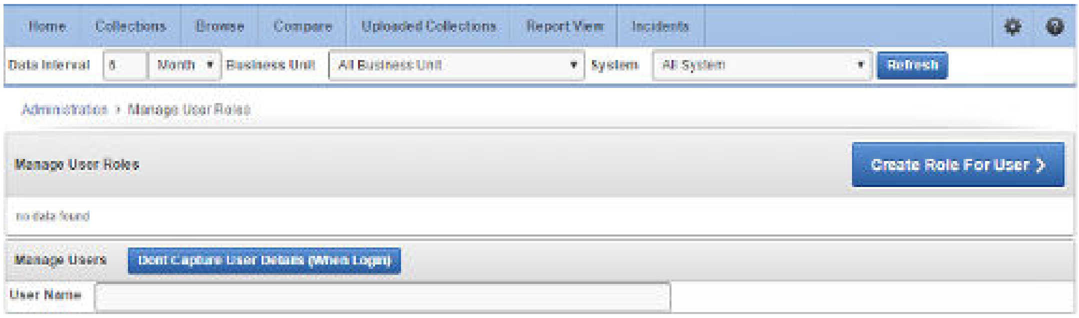
Manage Users, User Roles and assign System to users
1. Admin can Add/Delete Admins, DBA managers and assigning DBA's to DBA managers.
2. Admin can assigning Business Unit to DBA managers.
3. One DBA/DBA manager can belong to more than one Business Unit. One Business Unit can have more than one dba/dba managers. DBA Manager will have access to whole Business Unit and DBA's will have access to only particular System of Business Unit.
4. DBA Managers can add/remove DBA's for the units they have access to.
5. DBA's doesn't have access to ACL System.

Delete data
Where you can delete data older than certain time

Manage Notifications
Where you can (un)register for notifications and manage notifications.
Please make sure that email server is configured. Please check with admin.

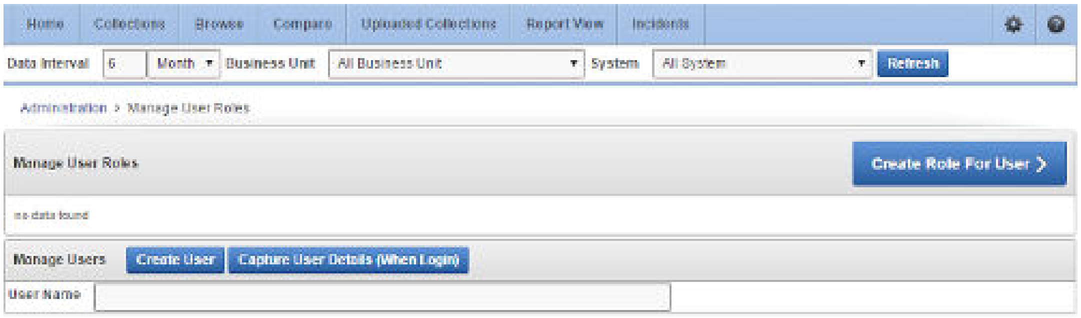
2. To disable automatic capture of users details, click **Don’t Capture User Details (When Login)**.

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)



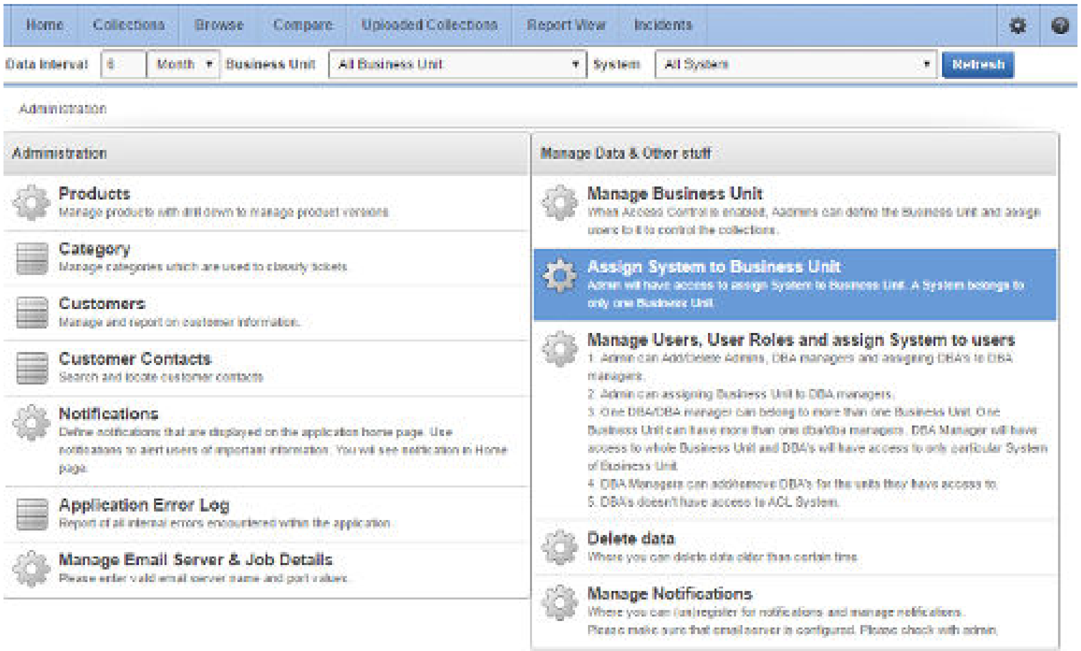
2.8.2 Bulk Mapping Systems to Business Units

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



Products
Manage products with drill down to manage product versions

Category
Manage categories which are used to classify tickets

Customers
Manage and report on customer information.

Customer Contacts
Search and locate customer contacts

Notifications
Define notifications that are displayed on the application home page. Use notifications to alert users of important information. You will see notification in Home page.

Application Error Log
Report of all internal errors encountered within the application

Manage Email Server & Job Details
Please enter valid email server name and port values.

Manage Business Unit
When Access Control is enabled, Admins can define the Business Unit and assign users to it to control the collections.

Assign System to Business Unit
Admin will have access to assign System to Business Unit. A System belongs to only one Business Unit.

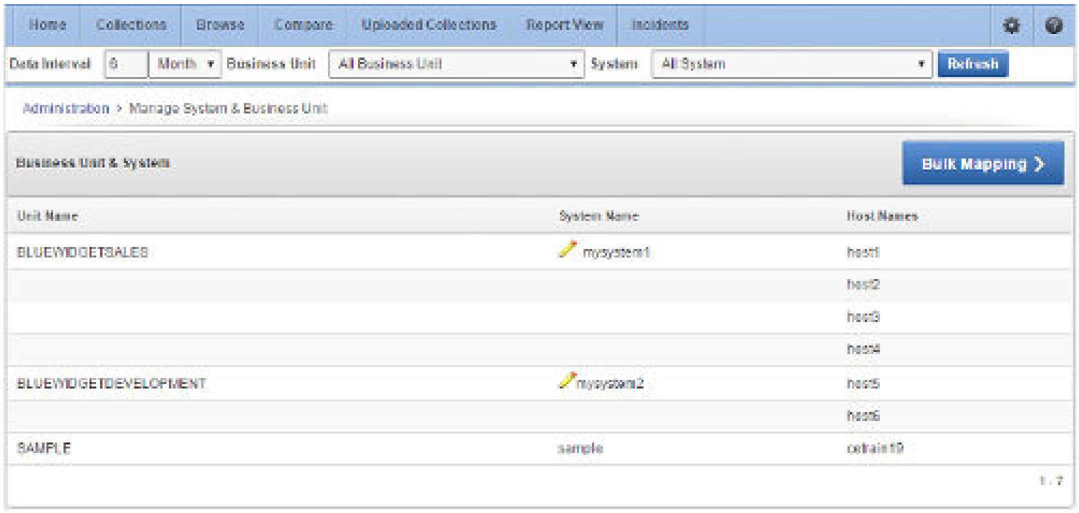
Manage Users, User Roles and assign System to users
1. Admin can Add/Delete Admins, DBA managers and assigning DBA's to DBA managers.
2. Admin can assigning Business Unit to DBA managers.
3. One DBA/DBA manager can belong to more than one Business Unit. One Business Unit can have more than one dba/dba managers. DBA Manager will have access to whole Business Unit and DBA's will have access to only particular System of Business Unit.
4. DBA Managers can add/remove DBA's for the units they have access to.
5. DBA's doesn't have access to ACL System.

Delete data
Where you can delete data older than certain time.

Manage Notifications
Where you can (un)register for notifications and manage notifications.
Please make sure that email server is configured. Please check with admin.

2. Click **Bulk Mapping**.

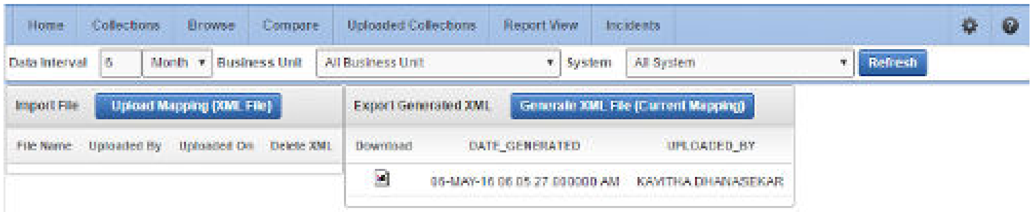
Figure 2-12 Bulk Mapping



Unit Name	System Name	Host Names
BLUEWIDGETSALES	mysystem1	host1
		host2
		host3
		host4
BLUEWIDGETDEVELOPMENT	mysystem2	host5
		host6
SAMPLE	sample	cetrain19

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

Figure 2-13 Upload a mapping XML



File Name	Uploaded By	Uploaded On	Delete XML

Download	DATE_GENERATED	UPLOADED_BY
	06-MAY-16 06.05.27.000000 AM	KAVITHA DHANASEKAR

   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

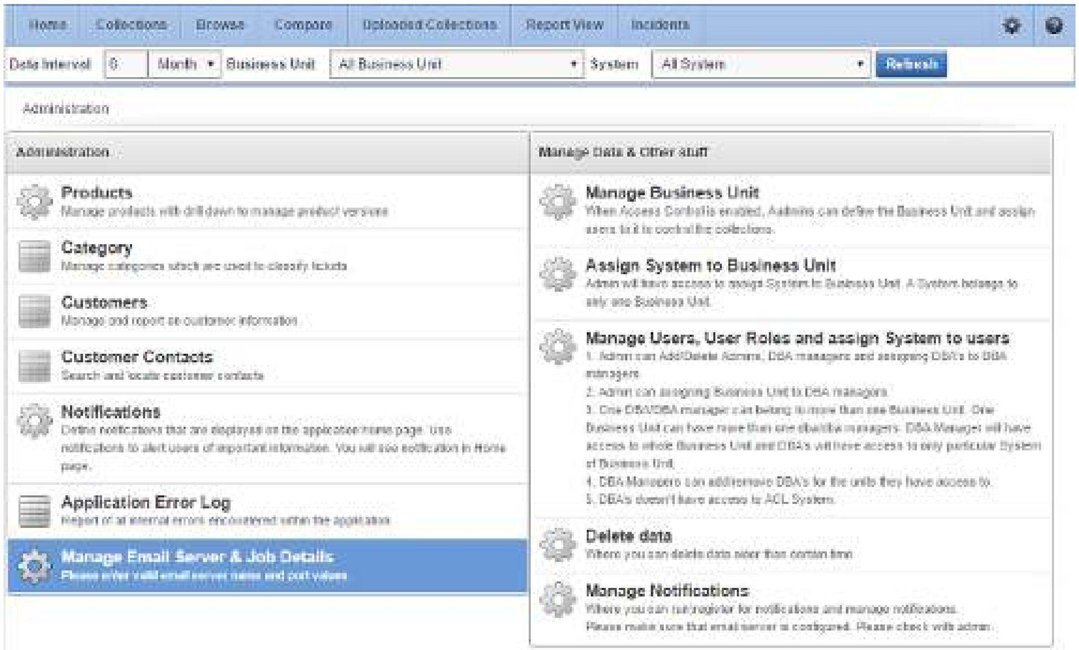
Modify or disable the purge schedule for Oracle Health Check Collections Manager collection data.

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



Products
Manage products with drill down to manage product versions

Category
Manage categories which are used to classify tickets

Customers
Manage and report on customer information

Customer Contacts
Search and locate customer contacts

Notifications
Define notifications that are displayed on the application home page. Use notifications to alert users of important information. You will see notification in Home page.

Application Error Log
Report of all internal errors encountered within the application

Manage Email Server & Job Details
Please enter valid email server name and port values

Manage Business Unit
When Access Control is enabled, Admins can define the Business Unit and assign users to it to control the collections.

Assign System to Business Unit
Admin will have access to assign System to Business Unit. A System belongs to only one Business Unit.

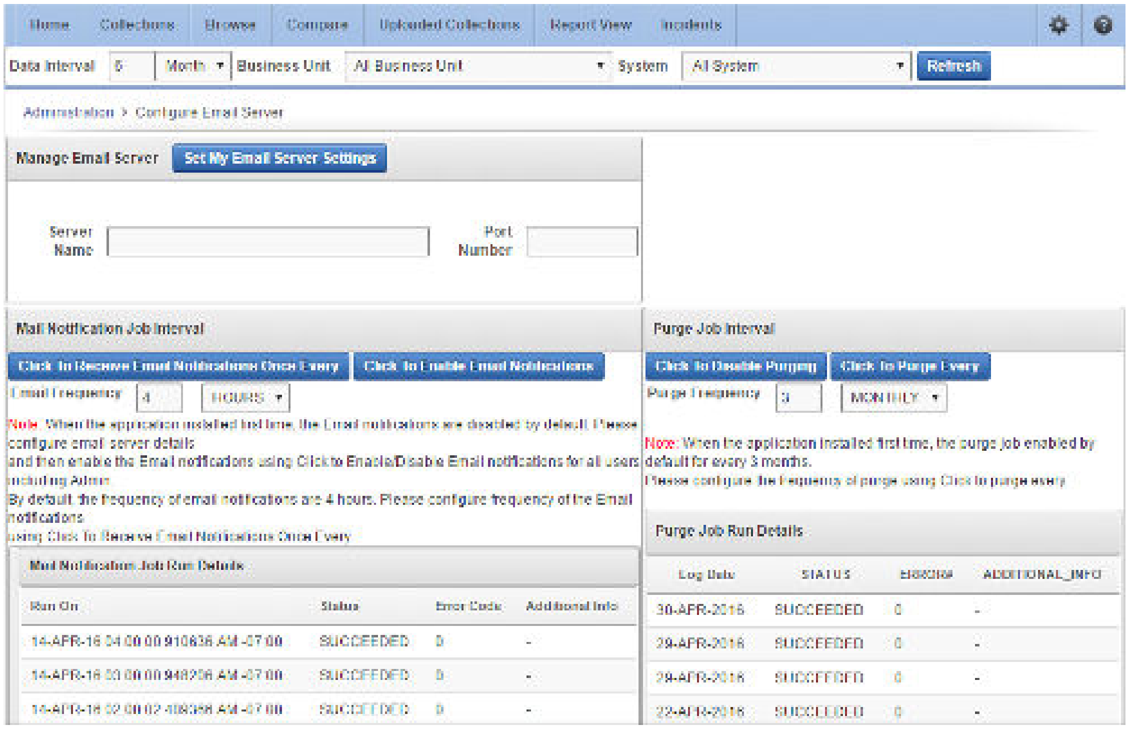
Manage Users, User Roles and assign System to users
1. Admin can Add/Delete Admins, DBA managers and assigning DBA's to DBA managers.
2. Admin can assigning Business Unit to DBA managers.
3. One DBA/DBA manager can belong to more than one Business Unit. One Business Unit can have more than one dba/dba managers. DBA Manager will have access to whole Business Unit and DBA's will have access to only particular System of Business Unit.
4. DBA Managers can add/remove DBA's for the units they have access to.
5. DBA's doesn't have access to ACL System.

Delete data
Where you can delete data older than certain time.

Manage Notifications
Where you can (un)register for notifications and manage notifications.
Please make sure that email server is configured. Please check with admin.

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


Home
Collections
Browse
Compare
Uploaded Collections
Report View
Incidents
Refresh
Administration > Configure Email Server
Manage Email Server
Set My Email Server Settings
Click To Receive Email Notifications Once Every
Click To Enable Email Notifications
Click To Disable Purging
Click To Purge Every

Data Interval 6 Month Business Unit All Business Unit System All System

Server Name

Port Number

Mail Notification Job Interval

Email Frequency 4 HOURS

Note: When the application installed first time, the Email notifications are disabled by default. Please configure email server details
and then enable the Email notifications using Click to Enable/Disable Email notifications for all users including Admin
By default the frequency of email notifications are 4 hours. Please configure frequency of the Email notifications
using Click To Receive Email Notifications Once Every

Mail Notification Job Run Details

Run On	Status	Error Code	Additional Info
14-APR-16 04.00.00.910636 AM -07:00	SUCCEEDED	0	-
14-APR-16 03.00.00.948256 AM -07:00	SUCCEEDED	0	-
14-APR-16 02.00.02.408358 AM -07:00	SUCCEEDED	0	-

Purge Job Interval

Purge Frequency 3 MONTHLY

Note: When the application installed first time, the purge job enabled by default for every 3 months.
Please configure the frequency of purge using Click to purge every

Purge Job Run Details

Log Date	STATUS	ERROR#	ADDITIONAL_INFO
30-APR-2016	SUCCEEDED	0	-
29-APR-2016	SUCCEEDED	0	-
29-APR-2016	SUCCEEDED	0	-
22-APR-2016	SUCCEEDED	0	-

2.8.4 Uploading Collections Automatically

Configure Oracle Autonomous Health Framework 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 Autonomous Health Framework 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 Oracle Autonomous Health Framework automatically uses the default values set in the `RAT_UPLOAD_USER` and `RAT_ZIP_UPLOAD_TABLE` environment variables.

3. Verify, using the `-checkdbupload` option if Oracle Autonomous Health Framework successfully connects to the database.

   ```
   $ orachk -checkdbupload
   ```

   ```
   $ exachk -checkdbupload
   ```

4. Set database uploads for Oracle Autonomous Health Framework check results.

   ```
   $ orachk -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 Autonomous Health Framework database check result uploads.

   ```
   $ orachk -getdbupload
   ```

Example 2-1 Checking Oracle Autonomous Health Framework Check Result Uploads

```
$ orachk -checkdbupload
Configuration is good to upload result to database.
```

At the end of health check collection, Oracle Autonomous Health Framework checks if the required connection details are set (in the wallet or the environment variables). If the connection details are set properly, then Oracle Autonomous Health Framework uploads the collection results.

To configure many Oracle Autonomous Health Framework 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 Autonomous Health Framework 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 Autonomous Health Framework 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.

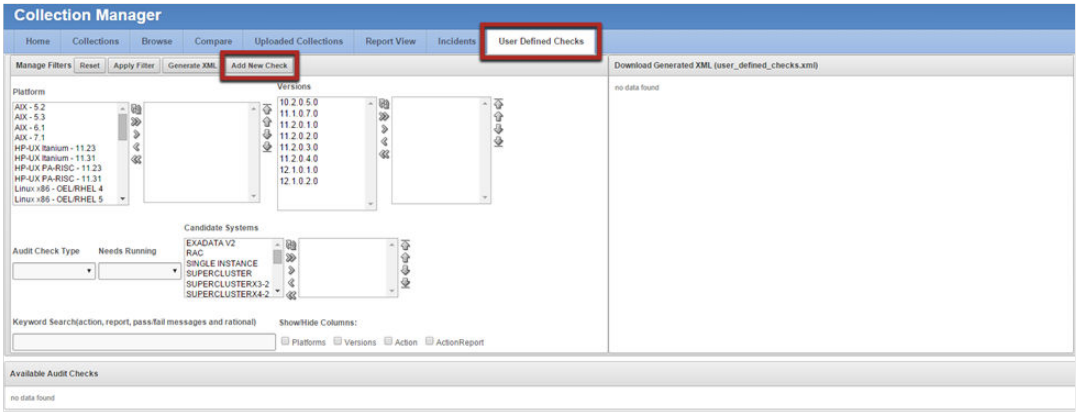
Oracle supports the framework for creating and running user-defined checks, but not the logic of the checks. It is your responsibility to test, verify, author, maintain, and support user-defined checks. At runtime, Oracle ORAchk and Oracle EXAchk script run the user-defined checks and display the results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

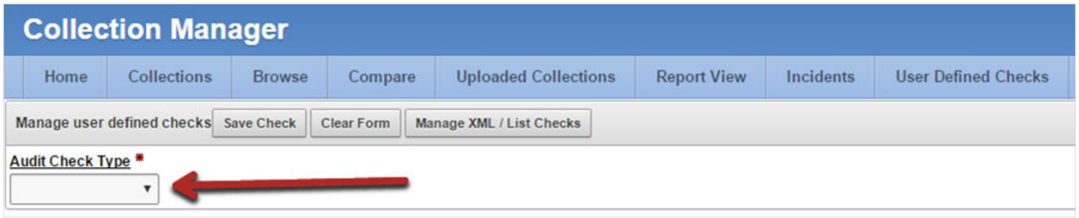
Figure 2-16 User-Defined Checks Tab



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

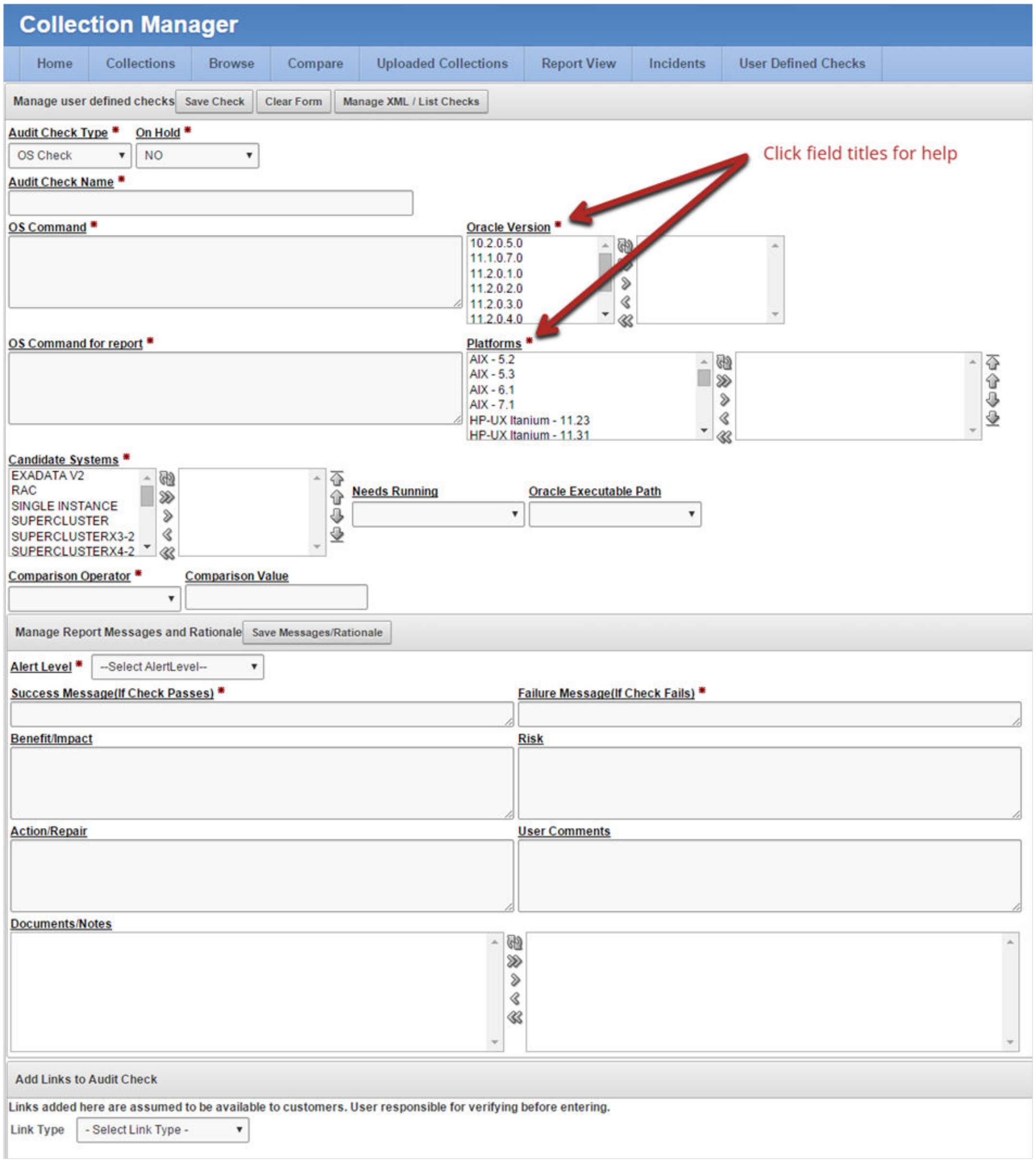


Once you have selected an **Audit Check Type**, Oracle Health Check Collections Manager updates the applicable fields.

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



Click field titles for help

Audit Check Type *: OS Check
On Hold *: NO
Audit Check Name *
OS Command *
Oracle Version *: 10.2.0.5.0, 11.1.0.7.0, 11.2.0.1.0, 11.2.0.2.0, 11.2.0.3.0, 11.2.0.4.0
OS Command for report *
Platforms *: AIX - 5.2, AIX - 5.3, AIX - 6.1, AIX - 7.1, HP-UX Itanium - 11.23, HP-UX Itanium - 11.31
Candidate Systems *: EXADATA V2, RAC, SINGLE INSTANCE, SUPERCLUSTER, SUPERCLUSTERX3-2, SUPERCLUSTERX4-2
Needs Running
Oracle Executable Path
Comparison Operator *
Comparison Value
Alert Level *: --Select AlertLevel--
Success Message(If Check Passes) *
Failure Message(If Check Fails) *
Benefit/Impact
Risk
Action/Repair
User Comments
Documents/Notes

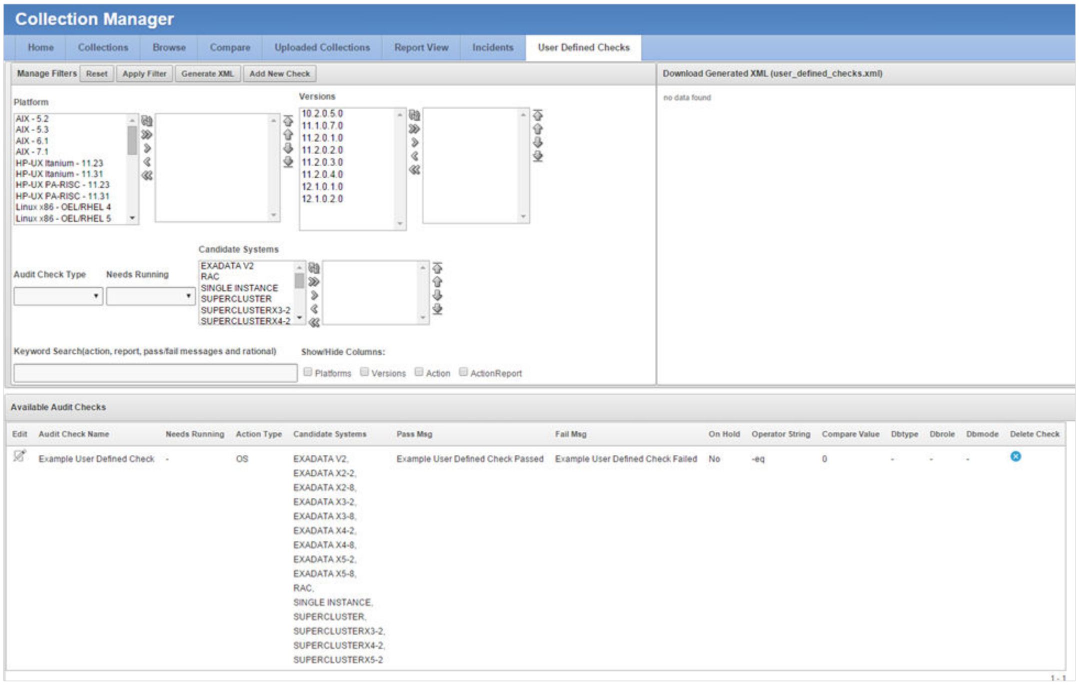
Links added here are assumed to be available to customers. User responsible for verifying before entering.

Link Type: - Select Link Type -

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



Platform:
AIX - 5.2
AIX - 5.3
AIX - 6.1
AIX - 7.1
HP-UX Itanium - 11.23
HP-UX Itanium - 11.31
HP-UX PA-RISC - 11.23
HP-UX PA-RISC - 11.31
Linux x86 - OEL/RHEL 4
Linux x86 - OEL/RHEL 5

Versions:
10.2.0.5.0
11.1.0.7.0
11.2.0.1.0
11.2.0.2.0
11.2.0.3.0
11.2.0.4.0
12.1.0.1.0
12.1.0.2.0

Candidate Systems:
EXADATA V2
RAC
SINGLE INSTANCE
SUPERCLUSTER
SUPERCLUSTERX3-2
SUPERCLUSTERX4-2

Download Generated XML (user_defined_checks.xml)

no data found

Edit	Audit Check Name	Needs Running	Action Type	Candidate Systems	Pass Msg	Fail Msg	On Hold	Operator String	Compare Value	Dbtype	Dbrole	Dbmode	Delete Check
	Example User Defined Check	-	OS	EXADATA V2, EXADATA X2-2, EXADATA X2-8, EXADATA X3-2, EXADATA X3-8, EXADATA X4-2, EXADATA X4-8, EXADATA X5-2, EXADATA X5-8, RAC, SINGLE INSTANCE, SUPERCLUSTER, SUPERCLUSTERX3-2, SUPERCLUSTERX4-2, SUPERCLUSTERX5-2	Example User Defined Check Passed	Example User Defined Check Failed	No	-eq	0	-	-	-	

3. Click the **Generate XML**.

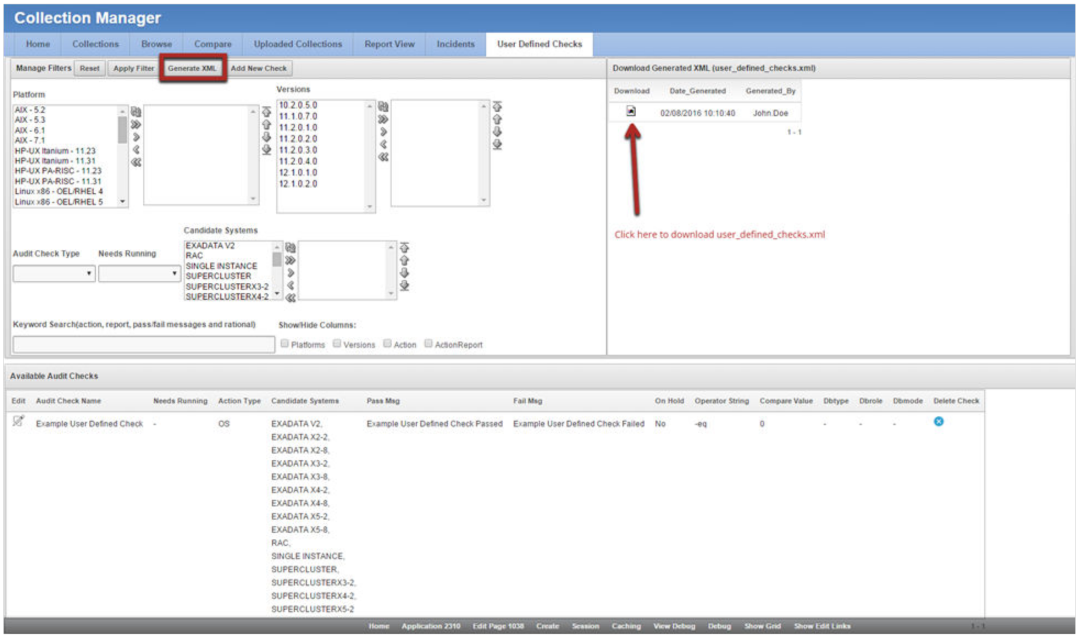
   On the right, find a link to download the generated `user_defined_checks.xml` file.

   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



Download	Date_Generated	Generated_By
	02/08/2016 10:10:40	John.Doe

1 - 1

Edit	Audit Check Name	Needs Running	Action Type	Candidate Systems	Pass Msg	Fail Msg	On Hold	Operator String	Compare Value	Dbtype	Dbrole	Dbmode	Delete Check
	Example User Defined Check	-	OS	EXADATA V2, EXADATA X2-2, EXADATA X2-8, EXADATA X3-2, EXADATA X3-8, EXADATA X4-2, EXADATA X4-8, EXADATA X5-2, EXADATA X5-8, RAC, SINGLE INSTANCE, SUPERCLUSTER, SUPERCLUSTERX3-2, SUPERCLUSTERX4-2, SUPERCLUSTERX5-2	Example User Defined Check Passed	Example User Defined Check Failed	No	-eq	0	-	-	-	

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 theUser 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`.

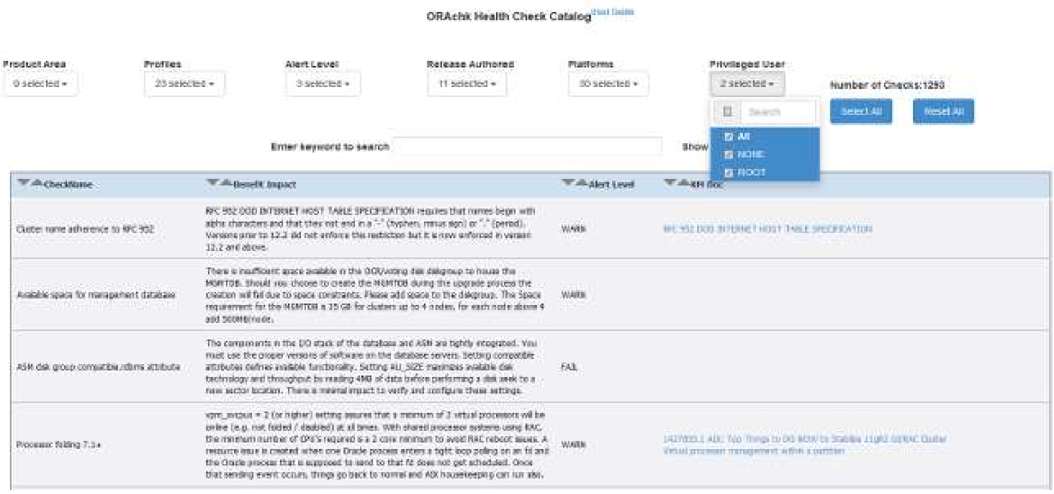
Enable Javascript before you view the Health Check Catalogs.

To filter health checks by privileged users:

1. Go to My Oracle Support note 2550798.1.
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 check boxes appropriately.

Figure 2-21 Oracle ORAchk - Privileged User



ORAchk Health Check Catalog User Guide

Product Area: 0 selected
Profiles: 23 selected
Alert Level: 3 selected
Release Authored: 11 selected
Platforms: 30 selected
Privileged User: 2 selected

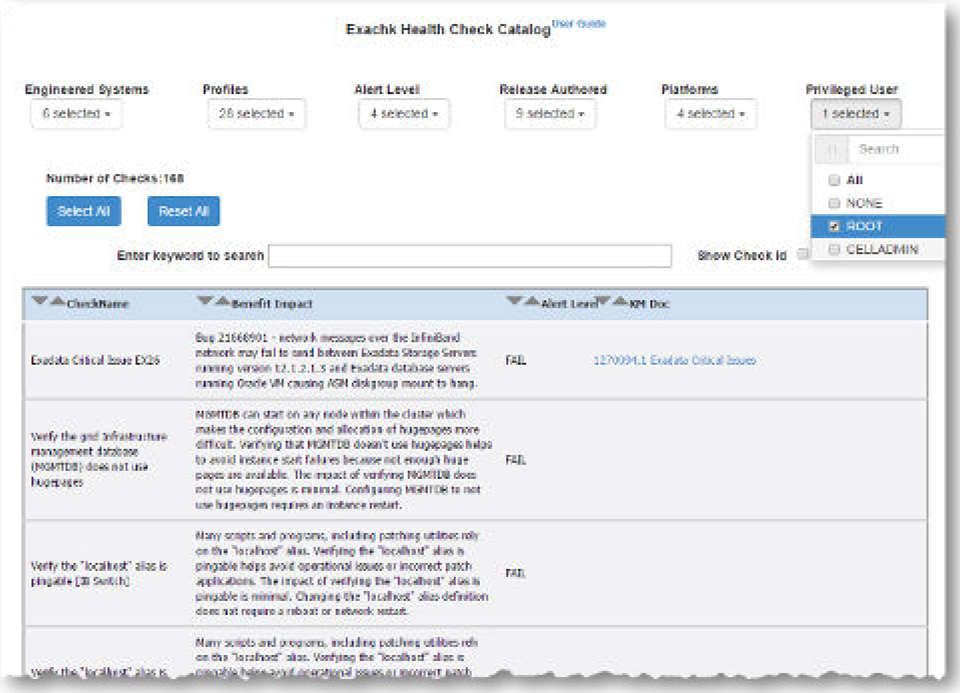
☑ All
☑ NONE
☑ ROOT

Number of Checks:1293

Enter keyword to search

CheckName	Benefit Impact	Alert Level	KM Doc
Cluster name adherence to RFC 952	RFC 952 DOD INTERNET HOST TABLE SPECIFICATION requires that names begin with alpha characters and that they not end in a "-" (hyphen, minus sign) or "." (period). Versions prior to 12.2 did not enforce this restriction but it is now enforced in version 12.2 and above.	WARN	RFC 952 DOD INTERNET HOST TABLE SPECIFICATION
Available space for management database	There is insufficient space available in the OCR/voting disk diskgroup to house the MGMTDB. Should you choose to create the MGMTDB during the upgrade process the creation will fail due to space constraints. Please add space to the diskgroup. The Space requirement for the MGMTDB is 35 GB for clusters up to 4 nodes, for each node above 4 add 500MB/node.	WARN	
ASM disk group compatible.rdbms attribute	The components in the I/O stack of the database and ASM are tightly integrated. You must use the proper versions of software on the database servers. Setting compatible attributes defines available functionality. Setting AU_SIZE maximizes available disk technology and throughput by reading 4MB of data before performing a disk seek to a new sector location. There is minimal impact to verify and configure these settings.	FAIL	
Processor folding 7.1+	vpm_xvcpus = 2 (or higher) setting assures that a minimum of 2 virtual processors will be online (e.g. not folded / disabled) at all times. With shared processor systems using RAC, the minimum number of CPU's required is a 2 core minimum to avoid RAC reboot issues. A resource issue is created when one Oracle process enters a tight loop polling on an fd and the Oracle process that is supposed to send to that fd does not get scheduled. Once that sending event occurs, things go back to normal and AIX housekeeping can run also.	WARN	1427855.1 AIX: Top Things to DO NOW to Stabilize 11gR2 GI/RAC Cluster Virtual processor management within a partition

Figure 2-22 Oracle EXAchk - Privileged User



Exachk Health Check Catalog User Guide

Engineered Systems: 6 selected
Profiles: 28 selected
Alert Level: 4 selected
Release Authored: 9 selected
Platforms: 4 selected
Privileged User: 1 selected

☐ All
☐ NONE
☑ ROOT
☐ CELLADMIN

Number of Checks:168

Enter keyword to search

Show Check Id ☐

CheckName	Benefit Impact	Alert Level	KM Doc
Exadata Critical Issue EX26	Bug 21666901 - network messages over the InfiniBand network may fail to send between Exadata Storage Servers running version 12.1.2.1.3 and Exadata database servers running Oracle VM causing ASM diskgroup mount to hang.	FAIL	1270094.1 Exadata Critical Issues
Verify the grid infrastructure management database (MGMTDB) does not use hugepages	MGMTDB can start on any node within the cluster which makes the configuration and allocation of hugepages more difficult. Verifying that MGMTDB doesn't use hugepages helps to avoid instance start failures because not enough huge pages are available. The impact of verifying MGMTDB does not use hugepages is minimal. Configuring MGMTDB to not use hugepages requires an instance restart.	FAIL	
Verify the "localhost" alias is pingable [IB Switch]	Many scripts and programs, including patching utilities rely on the "localhost" alias. Verifying the "localhost" alias is pingable helps avoid operational issues or incorrect patch applications. The impact of verifying the "localhost" alias is pingable is minimal. Changing the "localhost" alias definition does not require a reboot or network restart.	FAIL	
Verify the "localhost" alias is	Many scripts and programs, including patching utilities rely on the "localhost" alias. Verifying the "localhost" alias is pingable helps avoid operational issues or incorrect patch		

Related Topics

- https://support.oracle.com/rs?type=doc&id=2550798.1

2.8.8 Creating or Editing Incidents Tickets

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

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

- Creating Incident Tickets
- Editing Incident Tickets

2.8.8.1 Creating Incident Tickets

To create incident tickets:

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

Editing Incident Tickets

To edit incident tickets:

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

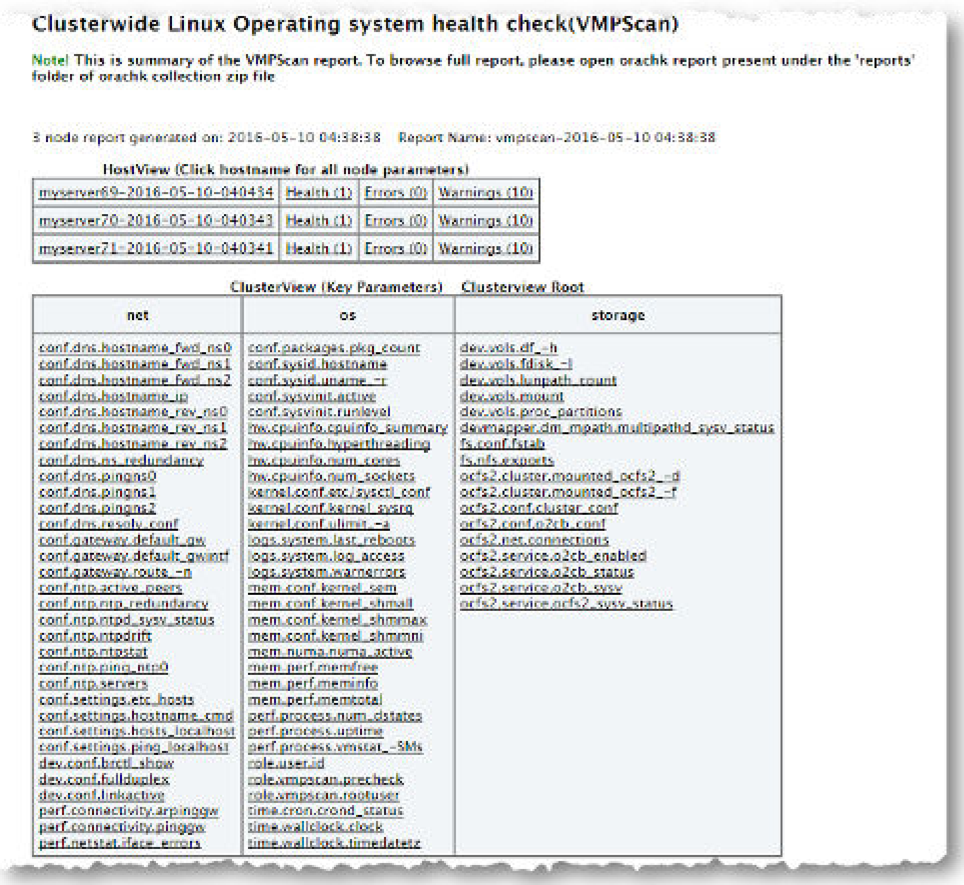
> **Note:**
>
> Click the delta symbol colored GREEN in the **Collections** or **Browse** tabs to edit incident tickets.

2.8.9 Viewing Clusterwide Linux Operating System Compliance Check (VMPScan)

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

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

Figure 2-23   Clusterwide Linux Operating System Health Check (VMPScan)



Clusterwide Linux Operating system health check(VMPScan)

Note! This is summary of the VMPScan report. To browse full report, please open orachk report present under the 'reports' folder of orachk collection zip file

3 node report generated on: 2016-05-10 04:38:38   Report Name: vmpscan-2016-05-10 04:38:38

HostView (Click hostname for all node parameters)

myserver69-2016-05-10-040434	Health (1)	Errors (0)	Warnings (10)
myserver70-2016-05-10-040343	Health (1)	Errors (0)	Warnings (10)
myserver71-2016-05-10-040341	Health (1)	Errors (0)	Warnings (10)

ClusterView (Key Parameters)   Clusterview Root

net	os	storage
conf.dns.hostname_fwd_ns0 conf.dns.hostname_fwd_ns1 conf.dns.hostname_fwd_ns2 conf.dns.hostname_ip conf.dns.hostname_rev_ns0 conf.dns.hostname_rev_ns1 conf.dns.hostname_rev_ns2 conf.dns.ns_redundancy conf.dns.pingns0 conf.dns.pingns1 conf.dns.pingns2 conf.dns.resolv_conf conf.gateway.default_gw conf.gateway.default_gwintf conf.gateway.route_-n conf.ntp.active_peers conf.ntp.ntp_redundancy conf.ntp.ntpd_sysv_status conf.ntp.ntpdrift conf.ntp.ntpstat conf.ntp.ping_ntp0 conf.ntp.servers conf.settings.etc_hosts conf.settings.hostname_cmd conf.settings.hosts_localhost conf.settings.ping_localhost dev.conf.brctl_show dev.conf.fullduplex dev.conf.linkactive perf.connectivity.arpinggw perf.connectivity.pinggw perf.netstat.iface_errors	conf.packages.pkg_count conf.sysid.hostname conf.sysid.uname_-r conf.sysvinit.active conf.sysvinit.runlevel hw.cpuinfo.cpuinfo_summary hw.cpuinfo.hyperthreading hw.cpuinfo.num_cores hw.cpuinfo.num_sockets kernel.conf.etc/sysctl_conf kernel.conf.kernel_sysrq kernel.conf.ulimit_-a logs.system.last_reboots logs.system.log_access logs.system.warnerrors mem.conf.kernel_sem mem.conf.kernel_shmall mem.conf.kernel_shmmax mem.conf.kernel_shmmni mem.numa.numa_active mem.perf.memfree mem.perf.meminfo mem.perf.memtotal perf.process.num_dstates perf.process.uptime perf.process.vmstat_-5Ms role.user.id role.vmpscan.precheck role.vmpscan.rootuser time.cron.crond_status time.wallclock.clock time.wallclock.timedatetz	dev.vols.df_-h dev.vols.fdisk_-l dev.vols.lunpath_count dev.vols.mount dev.vols.proc_partitions devmapper.dm_mpath.multipathd_sysv_status fs.conf.fstab fs.nfs.exports ocfs2.cluster.mounted_ocfs2_-d ocfs2.cluster.mounted_ocfs2_-f ocfs2.conf.cluster_conf ocfs2.conf.o2cb_conf ocfs2.net.connections ocfs2.service.o2cb_enabled ocfs2.service.o2cb_status ocfs2.service.o2cb_sysv ocfs2.service.ocfs2_sysv_status

Note:

The VMPScan report is included only when Oracle ORAchk is run on Linux systems.

2.9 Integrating Compliance Check Results with Other Tools

Integrate health check results from Oracle Autonomous Health Framework into Oracle Enterprise Manager and other third-party tools.

- Integrating Compliance Check Results with Oracle Enterprise Manager
  Integrate Oracle ORAchk and Oracle EXAchk compliance check results into Oracle Enterprise Manager.
- Integrating Compliance Check Results with Third-Party Tool
  Integrate compliance check results from Oracle Autonomous Health Framework into various third-party log monitoring and analytics tools, such as Elasticsearch and Kibana.

- Integrating Compliance Check Results with Custom Application
  Oracle Autonomous Health Framework uploads collection results from multiple instances into a single database for easier consumption of check results across your enterprise.

2.9.1 Integrating Compliance Check Results with Oracle Enterprise Manager

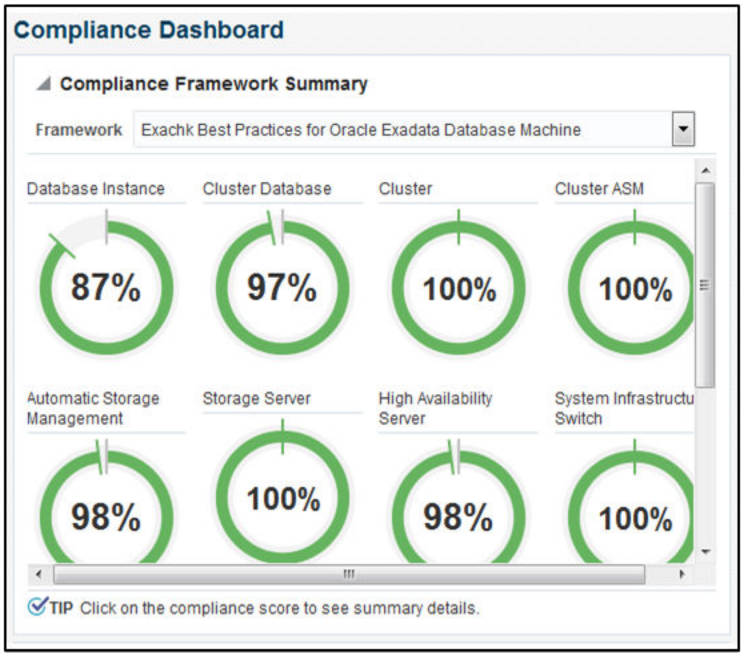
Integrate Oracle ORAchk and Oracle EXAchk compliance check results into Oracle Enterprise Manager.

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 Oracle ORAchk and Oracle EXAchk for Oracle Enterprise Manager Cloud Control 12*c* release 12.1.0.5 and earlier releases.

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

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

Figure 2-24 Compliance Dashboard



Compliance Dashboard

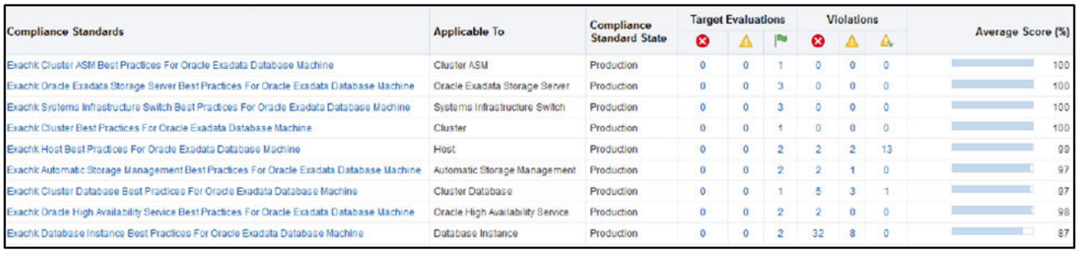
Compliance Framework Summary

Framework: Exachk Best Practices for Oracle Exadata Database Machine

TIP Click on the compliance score to see summary details.

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

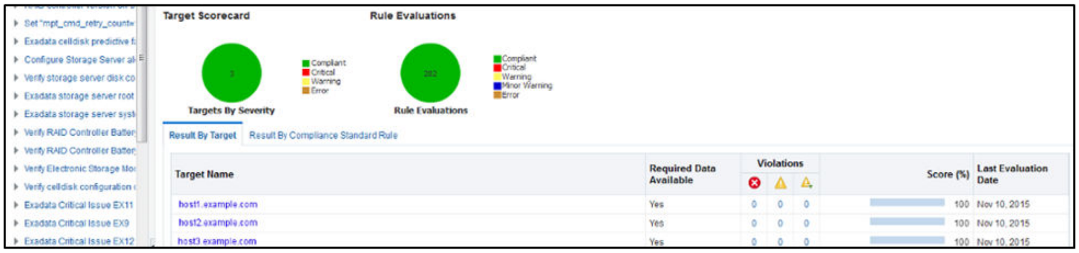
Figure 2-25 Compliance Standards



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



Set "mpt_cmd_retry_count=
Exadata celldisk predictive f
Configure Storage Server al
Verify storage server disk co
Exadata storage server root
Exadata storage server syst
Verify RAID Controller Batter
Verify RAID Controller Batter
Verify Electronic Storage Mo
Verify celldisk configuration
Exadata Critical Issue EX11
Exadata Critical Issue EX9
Exadata Critical Issue EX12
Target Scorecard
Rule Evaluations
3
Compliant
Critical
Warning
Error
Targets By Severity
282
Compliant
Critical
Warning
Minor Warning
Error
Rule Evaluations
Result By Target
Result By Compliance Standard Rule

Target Name	Required Data Available	Violations			Score (%)	Last Evaluation Date
		⊗	⚠	⚠		
host1.example.com	Yes	0	0	0	100	Nov 10, 2015
host2.example.com	Yes	0	0	0	100	Nov 10, 2015
host3.example.com	Yes	0	0	0	100	Nov 10, 2015

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

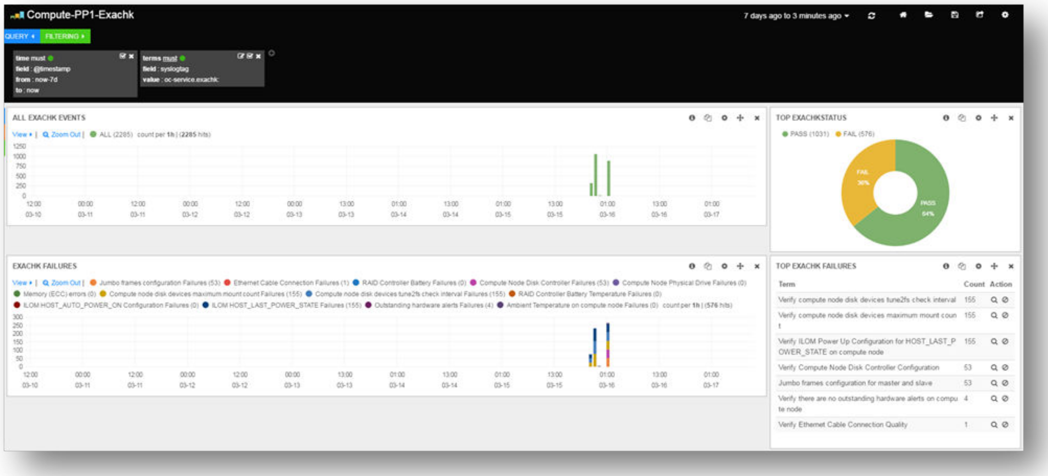
Related Topics

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

2.9.2 Integrating Compliance Check Results with Third-Party Tool

Integrate compliance check results from Oracle Autonomous Health Framework into various third-party log monitoring and analytics tools, such as Elasticsearch and Kibana.

Figure 2-27   Third-Party Tool Integration


Compute-PP1-Exachk
7 days ago to 3 minutes ago
QUERY
FILTERING
time must
field : @timestamp
from : now-7d
to : now
terms must
field : syslogtag
value : oc-service.exachk
ALL EXACHK EVENTS
View | Zoom Out | ● ALL (2285) count per 1h | (2285 hits)
1250
1000
750
500
250
0
12:00 03-10
00:00 03-11
12:00 03-11
00:00 03-12
12:00 03-12
00:00 03-13
13:00 03-13
01:00 03-14
13:00 03-14
01:00 03-15
13:00 03-15
01:00 03-16
13:00 03-16
01:00 03-17
TOP EXACHKSTATUS
● PASS (1031) ● FAIL (576)
FAIL 36%
PASS 64%
EXACHK FAILURES
View | Zoom Out | ● Jumbo frames configuration Failures (53) ● Ethernet Cable Connection Failures (1) ● RAID Controller Battery Failures (0) ● Compute Node Disk Controller Failures (53) ● Compute Node Physical Drive Failures (0) ● Memory (ECC) errors (0) ● Compute node disk devices maximum mount count Failures (155) ● Compute node disk devices tune2fs check interval Failures (155) ● RAID Controller Battery Temperature Failures (0) ● ILOM HOST_AUTO_POWER_ON Configuration Failures (0) ● ILOM HOST_LAST_POWER_STATE Failures (155) ● Outstanding hardware alerts Failures (4) ● Ambient Temperature on compute node Failures (0) count per 1h | (576 hits)
300
250
200
150
100
50
0
TOP EXACHK FAILURES

Term	Count	Action
Verify compute node disk devices tune2fs check interval	155	🔍 ⊘
Verify compute node disk devices maximum mount coun t	155	🔍 ⊘
Verify ILOM Power Up Configuration for HOST_LAST_P OWER_STATE on compute node	155	🔍 ⊘
Verify Compute Node Disk Controller Configuration	53	🔍 ⊘
Jumbo frames configuration for master and slave	53	🔍 ⊘
Verify there are no outstanding hardware alerts on compu te node	4	🔍 ⊘
Verify Ethernet Cable Connection Quality	1	🔍 ⊘

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 Autonomous Health Framework uses the message levels: `CRIT`, `ERR`, `WARN`, and `INFO`.

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

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

Related Topics

- Elasticsearch: RESTful, Distributed Search & Analytics | Elastic
- Kibana: Explore, Visualize, Discover Data | Elastic

- Logging Alerts to the syslogd Daemon

2.9.3 Integrating Compliance Check Results with Custom Application

Oracle Autonomous Health Framework uploads collection results from multiple instances into a single database for easier consumption of check results across your enterprise.

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

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

   Table 2-3   Uploading Collection Results into a Database

Table	What Get’s Uploaded
`rca13_docs`	Full zipped collection results.
`auditcheck_result`	Health check results.
`auditcheck_patch_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
  Configure Oracle Autonomous Health Framework to upload check results automatically to the Oracle Health Check Collections Manager database.

2.10 Using Oracle ORAchk to Confirm System Readiness for Implementing Application Continuity

Application Continuity Checking for Application Continuity enables you to deploy Application Continuity easily and transparently.

- Overview of Application Continuity
- Checks for Application Continuity
- Application Continuity Protection Check

2.10.1 Overview of Application Continuity

Oracle ORAchk identifies any references to deprecated Oracle JDBC concrete classes that need to be changed.

Oracle ORAchk analyzes the database operations in the application and reports the level of protection. It also reports where and why the applications are not protected.

Together, these checks can help you ensure that your application workload is covered by Oracle Application Continuity.

Related Topics

- http://docs.oracle.com/middleware/1213/wls/JDBCP/thirdparty.htm#JDBCP1028

2.10.2 Checks for Application Continuity

Application Continuity Checking for Concrete Classes

Determine whether Java applications use deprecated Oracle JDBC concrete classes.

To use Application Continuity with Java, replace the deprecated Oracle JDBC concrete classes. For information about the deprecation of concrete classes including actions to take if an application uses them, see My Oracle Support note 1364193.1.

To know if the application is using concrete classes, use Application Continuity checking (called `acchk` in Oracle ORAchk. Verify the application in advance while planning for high availability for your application.

For JDBC driver version 12.2.0.2 and below, Application Continuity is unable to replay transactions that use `oracle.sql` deprecated concrete classes of the form `ARRAY`, `BFILE`, `BLOB`, `CLOB`, `NCLOB`, `OPAQUE`, `REF`, or `STRUCT` as a variable type, a cast, the return type of a method, or calling a constructor.

For JDBC driver version 18c and above, Application Continuity is unable to replay transactions that use `oracle.sql` deprecated concrete classes of the form `OPAQUE`, `REF`, or `STRUCT` as a variable type, a cast, the return type of a method, or calling a constructor.

Modify them for Application Continuity to work with the application.

There are four values that control the Application Continuity checking for Oracle concrete classes. Set these values either on the command-line, or through shell environment variables, or mixed. The values are as follows:

Table 2-4 Application Continuity Checking for Concrete Classes

Command-Line Argument	Shell Environment Variable	Usage
`-asmhome jarfilename`	`RAT_AC_ASMJAR`	This must point to a version of `asm-all-5.0.3.jar` that you download from ASM Home page.
`-javahome JDK8dirname`	`RAT_JAVA_HOME`	This must point to the `JAVA_HOME` directory for a JDK8 installation.
`-appjar dirname`	`RAT_AC_JARDIR`	To analyze the application code for references to Oracle concrete classes, this must point to the parent directory name for the code. The program analyzes `.class` files, and recursively `.jar` files and directories.
`-jdbcver`	`RAT_AC_JDBCVER`	JDBC versions 12, 12.0.1, 18, 18.2, 19, or 19.1.

Example Application Continuity Concrete Class Checks Summary

The following command checks the Application Continuity checking for Oracle concrete classes.

```
$ orachk -acchk
-asmhome /path/orachk/asm-5.0.3/lib/all/asm-all-5.0.3.jar
-javahome /usr/lib/jvm/jre-1.8.0-openjdk.x86_64 -jdbcver 19.1
-appjar /scratch/nfs/tmp/jarfiles/app.jar


Outage Type      Status  Message
Concrete class checks   Total : 114 Passed : 110 Warning : 0 Failed : 4
(Failed check count is one per file)
                FAILED  [ac/workload/lobsanity/AnydaaOut]
[[CAST]desc=oracle/sql/ANYDATAmethodname=getDataInfo,lineno=38]
                FAILED  [ac/workload/lobsanity/AnydataOut]
[[CAST]desc=oracle/sql/ANYDATAmethodname=getDataInfo,lineno=38]
                FAILED  [ac/workload/lobsanity/AnydataSelect]
[[CAST]desc=oracle/sql/ANYDATAmethodname=queryAnydata,lineno=32]
                FAILED  [ac/workload/lobsanity/AnydataSelect]
[[CAST]desc=oracle/sql/ANYDATAmethodname=queryAnydata,lineno=32]
```

Application Continuity Checking for the Protection Level Application Continuity is Providing Your Application

Measure Coverage

Destructive testing is a good thing to do. However, introducing failures is non-deterministic. The application can fail over in all the tests, and then in production a failure occurs elsewhere and unexpectedly some requests do not fail over.

Using AC Check Coverage Analysis averts this situation by reporting in advance the percentage of requests that are fully protected by Application Continuity, and for the requests that are not fully protected, which they are and where. Use the coverage check before deployment, and after application changes. Developers and management know how to protect an application release from failures of the underlying infrastructure. If there is a problem, then it can be fixed before the application is released or waived knowing the level of coverage.

Executing the coverage check is rather like using SQL_TRACE. First run the application in a representative test environment with Application Continuity trace turned on at the server side. The trace is collected in the standard database user trace directory in user trace files. Then, pass this directory as input to Oracle ORAchk to report the coverage for the application functions. As this check uses Application Continuity, the database and client must be above 12c. The application need not necessarily released with Application Continuity. The check is to help you before release.

The following is a summary of the coverage analysis.

- If a round trip is made to the database server and returns while Application Continuity' capture is enabled during capture phase, then it is counted as a protected call.
- If a round trip is made to the database server while Application Continuity' capture is disabled (not in a request, or following a restricted call or a disable replay API was called), then it is counted as an unprotected call.
- Round trips ignored by capture and replay are ignored in the protection-level statistics.

At the end of processing each trace file, a level of protection for the calls sent to the database is computed.

For each trace: `PASS (>= 75)`, `WARNING (25 <= value <75)`, and `FAIL (< 25)`.

Running the Coverage Report

1. Turn on tracing at database level.
   Before running the workload, run the SQL statements as DBA on a test Oracle Database server so that the trace files include the needed information.

   For more details on setting or resetting Application Continuity trace events, please refer to sections Setting Application Continuity Trace Events Online in Memory and Setting Application Continuity Trace Events Offline in SPFILE.
2. Run through the application functions.
   To report on an application function, the application function must be run. The more application functions run, the better the information that the coverage analysis provides.
3. Use Oracle ORAchk to analyze the collected database traces and report the level of protection, and where not protected, reports why a request is not protected.
   To control the Application Continuity checking for coverage, set the following 2 values through command line or shell environment variables (or mixed):

Table 2-5 Using Application Continuity Checking for Protection Level

Command-Line Argument	Shell Environment Variable	Usage
`-javahome JDK8dirname`	`RAT_JAVA_HOME`	This must point to the `JAVA_HOME` directory for a JDK8 installation.
`-apptrc dirname`	`RAT_AC_TRCDIR`	To analyze the coverage, specify a directory name that contains one or more database server trace files. The trace directory is generally, `$ORACLE_BASE/diag/rdbms/{DB_UNIQUE_NAME}/$ORACLE_SID/trace`

Setting Application Continuity Trace Events Online in Memory

You can set or reset events in memory using the `alter system set events` command.

1. To turn on tracing online for all sessions in one instance:

   ```
   SQL> alter system set events
   'trace[progint_appcont_rdbms]:trace[sess_signature] disk highest:10602
   trace name context forever, level 28:10702 trace name context forever,
   level 16';
   ```

2. To turn off tracing online for all sessions in one instance:

   ```
   SQL> alter system set events 'trace[progint_appcont_rdbms] off:
   trace[sess_signature] off:10602
   trace name context off:10702 trace name context off';
   ```

3. To turn on tracing online per session:

   ```
   SQL> alter session set
   events='trace[progint_appcont_rdbms]:trace[sess_signature] disk
   highest:10602
   trace name context forever, level 28:10702 trace name context forever,
   level 16';
   ```

4. To turn off tracing online per session:

   ```
   SQL> alter system set events 'trace[progint_appcont_rdbms] off:
   trace[sess_signature] off:10602
   trace name context off:10702 trace name context off';
   ```

Setting Application Continuity Trace Events Offline in SPFILE

You cannot set or reset events in memory using the `alter system set event` command with `SCOPE` set to memory; instead, run the `alter system set event` command when you want to do changes to the SPFILE.

```
SQL> alter system set event='10602 trace name context forever, level
28:trace[progint_appcont_rdbms]:10702
trace name context forever, level 16';
```

```
System altered.
```

You need to restart the database instance after each change to the SPFILE.

Example Coverage Report

```
$ orachk -javahome /tmp/jdk1.8.0_40 -apptrc $ORACLE_BASE/diag/rdbms/
{DB_UNIQUE_NAME}/$ORACLE_SID/trace
```

Reading the Coverage Report

The coverage check produces a directory named `orachk_uname_date_time`. This report summaries coverage and lists trace files that have `WARNINGS` or `FAIL` status. To ensure all requests PASS (Coverage(%) = 100), check the PASS report, acchk_scorecard_pass.html. under the reports directory.

The output includes the database service name, the module name (from v$session.program, which can be set on the client side using the connection property on Java, for example, oracle.jdbc.v$session.program), the ACTION and CLIENT_ID, which can be set using setClientInfo with OCSID.ACTION and OCSID.CLIENTID respectively.

Example output: found in orachk_.....html#acchk_scorecard

```
Outage Type           Status  Message
Coverage checks               TotalRequest = 1088 PASS = 1082 WARNING = 1
FAIL = 5
                      FAIL    Trace file name = orcl1_ora_30467.trc Line
number of Request start = 1409 Request number = 6
                              SERVICE NAME = (srv_auto_pdb1) MODULE NAME =
(SQL*Plus) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 12 Protected Calls = 1
Unprotected Calls = 7
                      WARNING Trace file name = orcl1_ora_321597.trc Line
number of Request start = 653 Request number = 1
                              SERVICE NAME = (srv_ac_pdb2) MODULE NAME =
(JDBC Thin Client) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 25 Protected Calls = 1
Unprotected Calls = 3
                   FAIL    Trace file name = orcl1_ora_292714.trc Line number
of Request start = 1598 Request number = 7
                              SERVICE NAME = (srv_ac_pdb2) MODULE NAME =
(SQL*Plus) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 16 Protected Calls = 1
Unprotected Calls = 5
                      FAIL    Trace file name = orcl1_ora_112022.trc Line
number of Request start = 1167 Request number = 3
                              SERVICE NAME = (srv_ac_pdb2) MODULE NAME =
(JDBC Thin Client) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 0 Protected Calls = 0 Unprotected
Calls = 1
                      FAIL    Trace file name = orcl1_ora_112022.trc Line
number of Request start = 1353 Request number = 4
                              SERVICE NAME = (srv_ac_pdb2) MODULE NAME =
(JDBC Thin Client) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 0 Protected Calls = 0 Unprotected
Calls = 2
                      FAIL    Trace file name = orcl1_ora_112022.trc Line
number of Request start = 1689 Request number = 5
                              SERVICE NAME = (srv_ac_pdb2) MODULE NAME =
(JDBC Thin Client) ACTION NAME = () CLIENT ID = ()
                              Coverage(%) = 0 Protected Calls = 0 Unprotected
Calls = 1
                      PASS    Report containing checks that passed:
                              /scratch/nfs/orachk/
orachk_rwsbj14_060219_184513/reports/acchk_scorecard_pass.html
```

Related Topics

- Oracle Database Database Administrator’s Guide

- https://support.oracle.com/rs?type=doc&id=1364193.1
- ASM - Home Page

2.10.3 Application Continuity Protection Check

The Application Continuity Protection Check (ACCHK) feature generates Application Continuity coverage report for your applications.

For more information, see *Application Continuity Protection Check* in the *Oracle Real Application Clusters Administration and Deployment Guide*.

Related Topics

- Application Continuity Protection Check

2.11 Configuring Oracle REST Data Services (ORDS)

> **Note:**
>
> In the current release, REST services are not supported on Microsoft Windows.

- Configuring REST Using the Included ORDS
  Override default ORDS configuration by setting the shell environment variables.
- Configuring REST Using an Existing ORDS Installation

2.11.1 Configuring REST Using the Included ORDS

Override default ORDS configuration by setting the shell environment variables.

- By default, Oracle REST Data Services (ORDS) uses whichever port is available in the range `7080`-`7085`. If no port in this range is available, then ORDS exits and prompts you to set the `RAT_ORDS_PORT` environment variable. If `RAT_ORDS_PORT` is already set, then ORDS uses the port specified in the `RAT_ORDS_PORT` environment variable.
- By default, ORDS is setup with the administrator user `ordsadmin`. You can override this by specifying a different user in the `RAT_ORDSADMIN_USER` environment variable.
- Depending on Oracle ORAchk and Oracle EXAchk, ORDS is started as a `nologin` user named either `ordsorachk` or `ordsexachk`. If you use the ORDS, which is already running, then the user is as same as who is running ORDS.
- If Oracle Trace File Analyzer is installed, then ORDS picks `JAVA_HOME` from `TFA_HOME`. If Oracle Trace File Analyzer is not installed, then ORDS picks the default `JAVA_HOME`. It is a requirement that you use JDK8. However, you can override by setting the `RAT_JAVAEXE` environment variable.

2.11.2 Configuring REST Using an Existing ORDS Installation

1. To add the `oracHk.jar` file to the existing `ords.war` file:

   ```
   orachk -ordssetup ords_war_dir -configdir config_dir
   ```

   ```
   exachk -ordssetup ords_war_dir -configdir config_dir
   ```

   where,

   `ords_war_dir` is the directory that contains the `ords.war` file

   `config_dir` is an optional directory that you can specify to store the ORDS configuration files. If you do not specify the optional directory, then the configuration files are stored in the *`orda_war_dir`* directory.

   Stopping and restarting ORDS after running the `-ordssetup` command:

   - Adds the `orachk.jar` file to the existing `ords.war` file
   - Adds the user `ordsadmin` to the `ords.war` file, and grants `ORAchk admin` privileges to `ordsadmin`

2. To start the Oracle ORAchk or Oracle EXAchk daemon:

   ```
   orachk -d start -ords ords_war_dir
   ```

   ```
   exachk -d start -ords ords_war_dir
   ```

After completion, open the *`ords_war_dir`*`/log/ords_setup.log` file to view the REST URL details.

2.12 Using Oracle Autonomous Health Framework Compliance Over REST

Oracle ORAchk and Oracle EXAchk include full REST support allowing invocation and query over HTTPS.

- check
  Use GET requests to run a health check run for the specified check IDs.
- checktfafaileduploads
  Use GET requests to report if any Oracle Autonomous Health Framework service uploads failed.
- checktfaupload
  Use GET requests to report if a connection can be made to upload to Oracle Autonomous Health Framework service.
- download
  Use GET requests to download the collection result for the specified job ID.
- getinfo
  Use GET requests to report the status of the specified job ID.
- listcollections
  Use GET requests to get the list of Oracle ORAchk collections generated through REST.

- gettfaupload
  Use GET requests to report the Oracle Autonomous Health Framework service upload settings.
- profile
  Use GET requests to run a health check run for the specified profiles.
- showrepair
  Use GET requests to report the showrepair command for the specified check.
- start_client
  Use GET requests to run a normal health check run.
- start_client
  Use POST and GET requests to run a normal health check run using specific arguments.
- start_client
  Use POST requests to run a diff of the specified collection results.
- status
  Use GET requests to report the status on the specified job ID.
- unsettfaupload
  Use GET requests to unset all of the Oracle Autonomous Health Framework service upload settings, or a particular setting.
- uploadtfafailed
  Use GET requests to reattempt to upload all previously failed uploads to Oracle Autonomous Health Framework service.
- version
  Use GET requests to run the version command.

2.12.1 check

Use GET requests to run a health check run for the specified check IDs.

Syntax

```
/check/{check_id1,check_id2}
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"B2PKK9RR9M7MYJPRN8", "Status":"SUBMITTED" }]
```

Usage Notes

Specify a profile, or a comma-delimited list of check IDs.

Example 2-2 check

```
-bash-4.2$ curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/check/
E94589BC1AC24CFBE04312C0E50A3849


[{"ID":"B2PKK9RR9M7MYJPRN8","Status":"SUBMITTED"}]
```

2.12.2 checktfafaileduploads

Use GET requests to report if any Oracle Autonomous Health Framework service uploads failed.

Syntax

```
/checktfafaileduploads
```

Returns

If no collection failed to upload, then returns:

```
[{ "Msg":"There are no Failed collections under ORDS directory." }]
```

Or, prints the list of collections that failed to upload.

Usage Notes

You need not provide input to use this API.

Example 2-3 checktfafaileduploads

```
bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/checktfafaileduploads
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:04:58 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

[{"Msg":"There are no Failed collections under ORDS directory."}]
```

2.12.3 checktfaupload

Use GET requests to report if a connection can be made to upload to Oracle Autonomous Health Framework service.

Syntax

```
/checktfaupload
```

Returns

Returns JSON similar to:

```
[{ "ID":"ZFZLH06WOLE3L92PQI", "Status":"SUBMITTED" }]
```

Usage Notes

Use the `status` API to query the status of the submitted job.

Use the getinfo API to view the Oracle Autonomous Health Framework upload status after the status of the submitted API is COMPLETED.

Example 2-4   getinfo

With getinfo, returns:

```
[{"Msg":"Environment is not set for uploading results to TFA."}]
```

2.12.4 download

Use GET requests to download the collection result for the specified job ID.

Syntax

```
/download/{job_id}
```

Returns

Returns the zip binary for the collection result.

Usage Notes

Specify the job ID for which you want to download the collection result.

If you specify a purged ID or an invalid ID, then the error message will be in the downloaded file.

Example 2-5   download

```
# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/download/0K5Y5MAX2SD5CPP6SH
-O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 54854    0 54854    0     0   220k      0 --:--:-- --:--:-- --:--:--  221k

# unzip -qo 0K5Y5MAX2SD5CPP6SH

# ls -l
-rw-r--r--. 1 root root 54854 Sep  3 03:30 0K5Y5MAX2SD5CPP6SH
drwxr-xr-x. 1 root root   288 Sep  3 03:30
orachk_node1_orcl2_test_090319_032952_0K5Y5MAX2SD5CPP6SH
```

2.12.5 getinfo

Use GET requests to report the status of the specified job ID.

Syntax

```
/getinfo/{job_id}
```

Returns

Returns JSON similar to if the ID does not exist:

```
[{ "Status":"Either the ID entered is invalid or the wallet has been
purged." }]
```

Or, returns the repair command if the ID exists.

Usage Notes

Specify the job ID for which you want to check the status.

Example 2-6   getinfo

```
-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/getinfo/FJELUT7XYM3AKOE1R4
HTTP/1.1 200 OK Date: Thu, 19 Jul 2018 10:15:34 GMT
Content-Type: text/html X-Frame-Options: SAMEORIGIN Transfer-Encoding: chunked
```

Repair Command:

```
alter database datafile '+DATAC1/RAC12C/DATAFILE/sysaux.314.936528199'
autoextend on maxsize unlimited;
```

2.12.6 listcollections

Use GET requests to get the list of Oracle ORAchk collections generated through REST.

Syntax

```
/tfactl/orachk/listcollections
```

Returns

Returns the list of Oracle ORAchk collections generated through REST.

Example 2-7   listcollections

```
# curl -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/listcollections | sed s/
\<BR\>/\\n/g
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   581    0   581    0     0   3906      0 --:--:-- --:--:-- --:--:--  3925
List of collections:
/u02/test-user/oracle.ahf/data/node1/tfa/rest/ords/
orachk_node1_orcl2_bill_test_090319_034049_BTGP96ZYH45P5LHB86.zip
/u02/test-user/oracle.ahf/data/node1/tfa/rest/ords/
orachk_node1_orcl2_bill_test_090319_032952_0K5Y5MAX2SD5CPP6SH.zip
/u02/test-user/oracle.ahf/data/node1/tfa/rest/ords/
orachk_node1_orcl2_bill_test_090319_034245_WH9UWZRN9PKPDNKZCL.zip
```

```
/u02/test-user/oracle.ahf/data/node1/tfa/rest/ords/
orachk_node1_orcl2_bill_test_090319_033349_L05Y28DSOTZ9N73HO0.zip
```

2.12.7 gettfaupload

Use GET requests to report the Oracle Autonomous Health Framework service upload settings.

Syntax

```
/gettfaupload
```

Returns

Lists the values of three environment variables: `RAT_TFA_URL`, `RAT_TFA_USER`, and `RAT_TFA_PASSWORD`.

Usage Notes

You need not provide input to use this API.

Example 2-8 gettfaupload

```
bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/gettfaupload
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:07:24 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

RAT_TFA_URL =  https://tfa.example.com/tfa/ws/orachk/
RAT_TFA_USER =  orachkadmin
RAT_TFA_PASSWORD = ********
```

After `unsettfaupload` API, use the `gettfaupload` API to recheck the values:

```
-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/gettfaupload
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:10:10 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

RAT_TFA_URL is not set in the wallet
RAT_TFA_USER is not set in the wallet
RAT_TFA_PASSWORD is not set in the wallet
```

2.12.8 profile

Use GET requests to run a health check run for the specified profiles.

Syntax

```
/profile/{profile1},{profile2}
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"DMBLMBTB2M2H1QCQIS", "Status":"SUBMITTED" }]
```

Usage Notes

Specify a profile, or a list of profiles delimited by forward slash (/).

Example 2-9 profile

```
-bash-4.2$ curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/profile/asm

HTTP/1.1 200 OK Date: Thu, 05 Apr 2018 10:50:00 GMT Content-Type: text/html X-
Frame-Options:
SAMEORIGIN Transfer-Encoding: chunked Server: Jetty(9.2.z-SNAPSHOT)
[{"ID":"DMBLMBTB2M2H1QCQIS","Status":"SUBMITTED"}]
```

2.12.9 showrepair

Use GET requests to report the showrepair command for the specified check.

Syntax

```
/showrepair/{check_id}
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"ZFZLH06WOLE3L92PQI", "Status":"SUBMITTED" }]
```

Usage Notes

Specify the check ID for which you want to report the showrepair command.

Example 2-10 showrepair

```
-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/showrepair/
9ECBA2152E92F6B1E040E50A1EC00DFB
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:13:54 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
```

```
[{"ID":"FJELUT7XYM3AKOE1R4","Status":"SUBMITTED"}]


-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/status/FJELUT7XYM3AKOE1R4
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:15:00 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

[{"Msg":"Status of FJELUT7XYM3AKOE1R4 is  COMPLETED"}]
```

2.12.10 start_client

Use GET requests to run a normal health check run.

Syntax

```
/start_client
```

Returns

Returns JSON showing the job ID similar to:

```
[{
"ID":"UCTW5MLN7O1V1HPG8U",
"Status":"SUBMITTED"
}]
```

Usage Notes

You need not provide input to use this API.

Example 2-11   start_client

```
-bash-4.2$ curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/start_client

HTTP/1.1 200 OK Date: Thu, 05 Apr 2018 11:53:14 GMT Content-Type: text/html X-
Frame-Options:
SAMEORIGIN Transfer-Encoding: chunked Server: Jetty(9.2.z-SNAPSHOT)
[{"ID":"UCTW5MLN7O1V1HPG8U","Status":"SUBMITTED"}]
```

2.12.11 start_client

Use POST and GET requests to run a normal health check run using specific arguments.

Syntax

```
/start_client
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"UCTW5MLN7O1V1HPG8U", "Status":"SUBMITTED" }]
```

Usage Notes

Specify any Oracle ORAchk or Oracle EXAchk arguments and their corresponding values.

Example 2-12   JSON input

```
[{
"-clusternodes":"busm1c1,busm1c2",
"-ibswitches":"busm1sw-ibs0,busm1sw-iba0,busm1sw-ibb0"
}]
```

Example 2-13   start_client

```
# curl -i -X POST -H "Content-Type: application/json" -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/start_client -d '[{"-
clusternodes":"busm1c1,busm1c2","-ibswitches":"busm1sw-ibs0,busm1sw-
iba0,busm1sw-ibb0"}]


# curl -X POST -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/start_client -d '{"-check":
"81586F6DEC0DB43CE053D398EB0AF1EA", "-showpass", ""}'
[{"ID":"0K5Y5MAX2SD5CPP6SH","Status":"SUBMITTED"}]


# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/check/
81586F6DEC0DB43CE053D398EB0AF1EA
[{"ID":"BTGP96ZYH45P5LHB86","Status":"SUBMITTED"}]


# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/profile/asm
[{"ID":"WH9UWZRN9PKPDNKZCL","Status":"SUBMITTED"}]
```

2.12.12 start_client

Use POST requests to run a diff of the specified collection results.

Syntax

```
/start_client
```

Returns

Returns JSON similar to:

```
[{ "ID":"ZFZLH06WOLE3L92PQI", "Status":"SUBMITTED" }]
```

The status API can be used to query the status of the submitted job ID. Then you can use the download API to download diff report using the same job ID.

Usage Notes

JSON input:

```
[{ "-diff":"collection_zip_1 collection_zip_2" }]
```

Example 2-14   start_client

```
-bash-4.2$ curl -i -X POST -H "Content-Type: application/json" -k -u
tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/start_client -d '[{"-
diff":"orachk_myhost69_apxcmupg_062118_025029_N1O498NX877LYO5FE3.zip
orachk_myhost69_apxcmupg_062118_030527_ICMOWECU1UKF0R0VTO.zip"}]'
```

2.12.13 status

Use GET requests to report the status on the specified job ID.

Syntax

```
/status/{job_id}
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "Status of DMBLMBTB2M2H1QCQIS is SUBMITTED" }]
```

The status moves from `SUBMITTED` to `RUNNING` to `COMPLETED`.

Usage Notes

Specify the job ID for which you want to find the status.

Example 2-15   status

```
# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/status/DMBLMBTB2M2H1QCQIS

HTTP/1.1 200 OK Date: Thu, 05 Apr 2018 10:51:16 GMT Content-Type: text/html X-
Frame-Options:
```

```
SAMEORIGIN Transfer-Encoding: chunked Server: Jetty(9.2.z-SNAPSHOT)
[{"Status of DMBLMBTB2M2H1QCQIS is SUBMITTED"}]


# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/status/0K5Y5MAX2SD5CPP6SH
[{"Msg":"Status of 0K5Y5MAX2SD5CPP6SH is RUNNING"}]


# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/status/0K5Y5MAX2SD5CPP6SH
[{"Msg":"Status of 0K5Y5MAX2SD5CPP6SH is COMPLETED"}]


# curl -X GET -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/status/0K5Y5MAX2SD5CPP6SH
[{"Msg":"Status of 0K5Y5MAX2SD5CPP6SH is COMPLETED"}]
```

2.12.14 unsettfaupload

Use GET requests to unset all of the Oracle Autonomous Health Framework service upload settings, or a particular setting.

Syntax

```
/unsettfaupload/all
/unsettfaupload/RAT_TFA_USER
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"ZFZLH06WOLE3L92PQI", "Status":"SUBMITTED" }]
```

Usage Notes

Specify `all` to unset all of the three environment variables, `RAT_TFA_URL`, `RAT_TFA_USER`, and `RAT_TFA_PASSWORD` or, just specify an environment variable to unset it.

Example 2-16 unsettfaupload

```
-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/unsettfaupload/all
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:08:30 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

[{"ID":"Z8P9DHA8VV3PUOVQTV","Status":"SUBMITTED"}]
```

2.12.15 uploadtfafailed

Use GET requests to reattempt to upload all previously failed uploads to Oracle Autonomous Health Framework service.

Syntax

```
/uploadtfafailed/all
```

Returns

Returns JSON showing the job ID similar to:

```
[{ "ID":"ZFZLH06WOLE3L92PQI", "Status":"SUBMITTED" }]
```

Usage Notes

You need not provide input to use this API.

Example 2-17 uploadtfafailed

```
-bash-4.1# curl -i -X GET -k -u tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/uploadtfafailed/all
HTTP/1.1 200 OK
Date: Thu, 19 Jul 2018 10:09:18 GMT
Content-Type: text/html
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked

[{"ID":"0B9004CKSYZNUZCYZD","Status":"SUBMITTED"}]
```

2.12.16 version

Use GET requests to run the version command.

Syntax

```
/tfactl/orachk/version
```

Returns

Runs the `version` command and returns the version of the Oracle ORAchk daemon.

Example 2-18 version

```
# curl -k --user tfarest:password
https://node1.example.com:9090/ords/tfactl/orachk/version
{"VERSION":"ORACHK  VERSION: 19.3.0_20190902"}
```

2.13 Command-Line Options to Generate Password Protected Collection zip Files

Use the list of commands to encrypt or decrypt diagnostic collection `zip` files.

Table 2-6 Encrypt and Decrypt Diagnostic Collection zip Files

Option	Description

`orachk -d start -encryptzip`
`exachk -d start -encryptzip`
	Starts the daemon with `-encryptzip` option.
	The daemon prompts for a password when it starts. The daemon then encrypts the subsequent on-demand and scheduled runs collections with that password.
	Note: When `-encryptzip` is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection `zip` file deletes the collections directory.

`orachk [-option *value*] -encryptzip`
`exachk [-option *value*] -encryptzip`
	Encrypts the run result.
	Prompts for the password, and encrypts the collections created at the end of the run with that password.
	You can use `-encryptzip` with other Oracle ORAchk and Oracle EXAchk options that generate a collection.
	For example:
	`orachk -profile *profile-name* -encryptzip`
	`orachk -profile sysadmin -encryptzip`
	`orachk -check *check-id* -encryptzip`
	`orachk -check D47661C55B1A291AE0431EC0E50A5C53 -encryptzip`
	Note: When `-encryptzip` is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection `zip` file deletes the collections directory.

Table 2-6 (Cont.) Encrypt and Decrypt Diagnostic Collection zip Files

Option	Description

`orachk -encryptzip` *`zip_file`*
`exachk -encryptzip` *`zip_file`*
	Encrypts the already generated collection.
	Prompts for the password, encrypts the zip file specified with that password, and then renames the collections as, for example, `orachk_`*`host_db`*`_encrypted_`*`date_time`*`.zip`.
	Note:
	When `-encryptzip` is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection `zip` file deletes the collections directory.

`orachk -decryptzip` *`zip_file`*
`exachk -decryptzip` *`zip_file`*
	Decrypts the encrypted collection.
	Prompts for the password, decrypts the `zip` file specified with that password, and then renames the collections as, for example, `orachk_`*`host_db_date_time`*`.zip`.

2.14 Caching Discovery Data

Use the list of commands to manage caching of discovery data.

Syntax

```
orachk -discovery -discoverydir location
exachk -discovery -discoverydir location
```

```
orachk -checkdiscovery
exachk -checkdiscovery
```

```
orachk -usediscovery -discoverydir location
exachk -usediscovery -discoverydir location
```

```
orachk -rediscovery
exachk -rediscovery
```

```
orachk -rmdiscovery
exachk -rmdiscovery
```

Table 2-7 Manage Caching of Discovery Data

Command	Description
`-discovery`	Caches discovery data, which Oracle ORAchk and Oracle EXAchk can use for future runs. `-discoverydir`: Specify the location to store the discovery data.
`-checkdiscovery`	Verifies the discovery data.
`-usediscovery`	Uses the discovery data. `-discoverydir`: Specify the location where you have cached the discovery data.
`-rediscovery`	Refreshes the cache discovery data.
`-rmdiscovery`	Removes the cached discovery data.

2.15 Applying Patch Between Releases

Use the list of commands to manage patches.

Syntax

```
orachk -applypatch orachk_bug_id.zip
exachk -applypatch exachk_bug_id.zip


orachk -querypatch all
exachk -querypatch all
orachk -querypatch bug_id
exachk -querypatch bug_id


orachk -rollbackpatch bug_id
exachk -rollbackpatch bug_id
```

Table 2-8 Managing Patches

Command	Description
`-applypatch`	Applies a new patch for the specified bug ID.
`-querypatch`	Lists the details of all of the installed patches or for the specified bug ID.
`-rollbackpatch`	Rolls back the applied patch to its previous state, the state at which the patch was applied.

2.16 Creating, Modifying, and Deleting User-Defined Profiles

Specify a comma-delimited list of check IDs to create and modify custom profiles.

Specify valid check IDs and descriptive unique profile name.

1. To create a profile:

   ```
   orachk -createprofile profile_name check_ids
   ```

   ```
   exachk -createprofile profile_name check_ids
   ```

   ```
   orachk -createprofile customprofile1 E94AC6ACDA502F3BE04312C0E50A290A,
   F01E3FEDBD2B243EE04312C0E50A4DC5,
   F02293F7261D1BCAE04312C0E50A4118,
   F9370B4F5707076DE04312C0E50A78AE

   Validating checks...

   Profile customprofile1 created successfully...
   ```

   Oracle ORAchk and Oracle EXAchk validate profile names and check IDs before creating the profile and print appropriate messages if any discrepancies found. Oracle ORAchk and Oracle EXAchk create the profiles only if the profile names are unique and check IDs are valid.

2. To modify a profile:

   ```
   orachk -modifyprofile profile_name check_ids
   ```

   ```
   exachk -modifyprofile profile_name check_ids
   ```

   ```
   exachk -modifyprofile customprofile1 21B57D4065DDEA3DE0530D98EB0A8205,
   39128FBB540C098AE0530D98EB0AFB1A,
   9AD8AF3966FB3027E040E50A1EC0308F,
   019F5085951978CAE05313C0E50A4FCB

   Validating checks...

   Modifying profile customprofile1...

   Profile customprofile1 modified successfully...


   Added Checks:
   21B57D4065DDEA3DE0530D98EB0A8205
   9AD8AF3966FB3027E040E50A1EC0308F
   019F5085951978CAE05313C0E50A4FCB
   --------------------------------
   Removed Checks:
   39128FBB540C098AE0530D98EB0AFB1A
   ```

   You cannot modify the profile name. You can only add to or remove check IDs form the profile.
   If the check IDs are in the profile, then Oracle ORAchk and Oracle EXAchk remove them from the profile.

If the check IDs are not in the profile, then Oracle ORAchk and Oracle EXAchk add them to the profile.

3. To delete a profile:

```
orachk -deleteprofile profile_name
```

```
exachk -deleteprofile profile_name
```

```
orachk -deleteprofile customprofile1

Deleting profile customprofile1...

Profile customprofile1 deleted successfully...
```

Oracle ORAchk and Oracle EXAchk delete the profile by removing the profile entry ID from the `profiles.dat` file, and deleting the corresponding `profiles.prf` file.

2.17 Sanitizing Sensitive Information in the Diagnostic Collections

Oracle Autonomous Health Framework uses Adaptive Classification and Redaction (ACR) to sanitize sensitive data.

After collecting copies of diagnostic data, Oracle ORAchk and Oracle EXAchk use Adaptive Classification and Redaction (ACR) to sanitize sensitive data in the collections. ACR uses a machine learning based engine to redact a pre-defined set of entity types in a given set of files. ACR also sanitizes or masks entities that occur in path names.

- Sanitization replaces a sensitive value with random characters.
- Masking replaces a sensitive value with a series of asterisks ("*").

ACR currently sanitizes the following entity types:

- Host names
- IP addresses
- MAC addresses
- Oracle Database names
- Tablespace names
- Service names
- Ports
- Operating system user names

ACR also masks Personally Identifiable Information (PII), that is, user data from the database appearing in block and redo dumps. There is no separate command for it.

To sanitize sensitive information:

```
orachk -sanitize comma_delimited_list_of_collection_IDs
```

or

```
exachk -sanitize comma_delimited_list_of_collection_IDs
```

Block dumps before redaction:

```
14A533F40 00000000 00000000 00000000 002C0000 [..............,.]
14A533F50 35360C02 30352E30 31322E37 380C3938 [..650.507.2189.8]
14A533F60 31203433 37203332 2C303133 360C0200 [34 123 7310,...6]
```

Block dumps after redaction:

```
14A533F40 ******** ******** ******** ******** [****************]
14A533F50 ******** ******** ******** ******** [****************]
14A533F60 ******** ******** ******** ******** [****************]
```

Redo dumps before redaction:

```
col 74: [ 1] 80
col 75: [ 5] c4 0b 19 01 1f
col 76: [ 7] 78 77 06 16 0c 2f 26
```

Redo dumps after redaction:

```
col 74: [ 1] **
col 75: [ 5] ** ** ** ** **
col 76: [ 7] ** ** ** ** ** ** **
```

To print the reverse map of sanitized elements:

```
orachk -rmap all|comma_delimited_list_of_element_IDs
```

or

```
exachk -rmap all|comma_delimited_list_of_element_IDs
```

Sanitizing Sensitive Information in Oracle ORAchk or Oracle EXAchk Output

1. If you specify a file name that does not follow the naming convention:

   For example:

   ```
   $ orachk -sanitize orachk_invalid.html
   /scratch/testuser/may31/orachk_invalid.html is not a valid orachk
   collection
   ```

2. If you specify a file that does not exist:

For example:

```
$ orachk -sanitize /tmp/orachk_invalid.html
/tmp/orachk_invalid.html does not exist
```

3. If you sanitize a file that exists with valid Oracle Autonomous Health Framework naming convention, but the file is not generated by Oracle Autonomous Health Framework:

   For example:

   ```
   $ orachk -sanitize orachk_invalidcollection.zip
   orachk is sanitizing /scratch/testuser/may31/orachk_invalidcollection.zip.
   Please
   wait...
   ACR error occurred while sanitizing orachk collection
   ```

4. To sanitize a file with relative path:

   For example:

   ```
   $ orachk -sanitize new/orachk_node061919_053119_001343.zip
   orachk is sanitizing
   /scratch/testuser/may31/new/orachk_node061919_053119_001343.zip. Please
   wait...

   Sanitized collection is:
   /scratch/testuser/may31/orachk_aydv061919_053119_001343.zip


   $ orachk -sanitize .orachk_node061919_053119_001343.zip
   orachk is sanitizing
   /scratch/testuser/may31/.orachk_node061919_053119_001343.zip. Please
   wait...

   Sanitized collection is:
   /scratch/testuser/may31/orachk_aydv061919_053119_001343.zip
   ```

5. To sanitize Oracle Autonomous Health Framework debug log:

   For example:

   ```
   $ orachk -sanitize new/orachk_debug_053119_023653.log
   orachk is sanitizing /scratch/testuser/may31/new/
   orachk_debug_053119_023653.log.
   Please wait...

   Sanitized collection is: /scratch/testuser/may31/
   orachk_debug_053119_023653.log
   ```

6. To run full sanity check:

   For example:

   ```
   $ orachk -localonly -profile asm -sanitize -silentforce

   Detailed report (html) -
   /scratch/testuser/may31/orachk_node061919_053119_04448/
   ```

```
orachk_node061919_053119_04448.html

orachk is sanitizing /scratch/testuser/may31/
orachk_node061919_053119_04448.
Please wait...

Sanitized collection is: /scratch/testuser/may31/
orachk_aydv061919_053119_04448

UPLOAD [if required] - /scratch/testuser/may31/
orachk_node061919_053119_04448.zip
```

7. To print the reverse map of sanitized elements:

   For example:

```
orachk -rmap pu406jKxg,kEvGFDT
___________________________________________________________________________
_____
| Entity Type | Substituted Entity Name | Original Entity Name |
___________________________________________________________________________
_____
| dbname      | XTT_MANUR               | ASM_POWER            |
| dbname      | fcb63u2                 | rac12c2              |
___________________________________________________________________________
_____


orachk -rmap all
```

2.18 Troubleshooting Compliance Framework (Oracle ORAchk and Oracle EXAchk)

Follow the steps explained in this section to troubleshoot and fix Compliance Framework (Oracle ORAchk / Oracle EXAchk) related issues.

- How to Troubleshoot Oracle ORAchk and Oracle EXAchk Issues
  Follow these steps to fix the Oracle ORAchk and Oracle EXAchk related issues.
- How to Capture Debug Output
  Follow these procedures to capture debug information.
- Remote Login Problems
  If Oracle ORAchk and Oracle EXAchk have problem locating and running SSH or SCP, then the tools cannot run any remote checks.
- Permission Problems
  You must have sufficient directory permissions to run Oracle ORAchk and Oracle EXAchk.
- Slow Performance, Skipped Checks, and Timeouts
  Follow these procedures to fix slow performance and other issues.

Related Topics

- *Oracle Autonomous Health Framework Checks and Diagnostics User's Guide*

2.18.1 How to Troubleshoot Oracle ORAchk and Oracle EXAchk Issues

Follow these steps to fix the Oracle ORAchk and Oracle EXAchk related issues.

1. Ensure that you are using the correct tool.

   If you have an Oracle Engineered System other than Oracle Database Appliance, then use Oracle EXAchk. For all other systems, use Oracle ORAchk.

2. Ensure that you are using the latest versions of Oracle ORAchk and Oracle EXAchk.

   New versions are released every three months.

   a. Check the version using the `-v` option.

      ```
      $ orachk -v
      ```

      ```
      $ exachk -v
      ```

   b. Compare your version with the latest version available here:

      i. For Oracle ORAchk, refer to My Oracle Support note 2550798.1.

      ii. For Oracle EXAchk, refer to My Oracle Support note 1070954.1.

3. Check the **FAQ** for similar problems in My Oracle Support note 1070954.1.

4. Review files within the `log` directory.

   a. Check applicable `error.log` files for relevant errors.

      This file contains `stderr` output captured during the run, not everything you see in here will mean you have a problem, but if you have a problem this may give more information.

      - `*output_dir*/log/orachk _error.log`
      - `*output_dir*/log/exachk _error.log`

   b. Check 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 My Oracle Support Community (MOSC).

7. If necessary capture debug output, log a new SR and attach the resulting `zip` file.

Related Topics

- https://support.oracle.com/rs?type=doc&id=2550798.1
- https://support.oracle.com/rs?type=doc&id=1070954.1
- My Oracle Support Community (MOSC)

2.18.2 How to Capture Debug Output

Follow these procedures to capture debug information.

1. Before enabling debug, reproduce the problem with the least run necessary.

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

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

2. Enable debug.

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

   If the problem area is known, then debug can be constrained to a particular module by including the `-module` argument too.

   ```
   $ orachk -debug [-module [ setup | discovery | execution | output ] ]
   ```

   ```
   $ exachk -debug [-module [ setup | discovery | execution | output ] ]
   ```

   When debug is enabled, Oracle ORAchk and Oracle EXAchk create a new debug log file in:

   - *output_dir*/log/orachk _debug_*date_stamp_time_stamp*.log
   - *output_dir*/log/exachk _debug_*date_stamp_time_stamp*.log

   The *output_dir* directory retains a number of other temporary files used during health checks.

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

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

   ```
   $ orachk -d start -debug
   ```

   ```
   $ exachk -d start -debug
   ```

   When debug is run with the daemon, Oracle ORAchk and Oracle EXAchk create a daemon debug log file in the directory 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.

2.18.3 Remote Login Problems

If Oracle ORAchk and Oracle EXAchk 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, `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 ...
```

2.18.4 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
```

- If Oracle Clusterware is installed, then:
  - Install Oracle EXAchk in `/opt/oracle.SupportTools/exachk` as the Oracle Grid Infrastructure home owner
  - Install Oracle ORAchk in `CRS_HOME/suptools/orachk` as the Oracle Grid Infrastructure home owner
- If Oracle Clusterware is not installed, then:
  - Install Oracle EXAchk in `/opt/oracle.SupportTools/exachk` as `root`
  - Install Oracle ORAchk (in a convenient location) as `root` (if possible)

    or

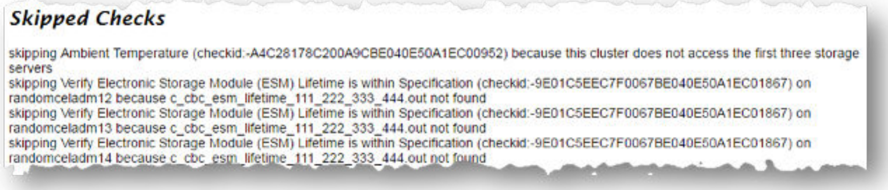
    Install Oracle ORAchk (in a convenient location) as Oracle software install user or Oracle Database home owner

2.18.5 Slow Performance, Skipped Checks, and Timeouts

Follow these procedures 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 2-28 Skipped Checks



Skipped Checks

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

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

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

1. Determine if there is a pattern to what is causing the problem.
   - EBS checks, for example, depend on the amount of data present and may take longer than the default timeout.
   - If there are prompts in the remote profile, then remote checks timeout and be killed and skipped. 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.

   - You override the default timeouts by setting the environment variables.

Table 2-9 Timeout Controlling

Timeout Controlling	Default Value (seconds)	Environment Variable
Collection of all checks not run by `root` (most). Specify the timeout value for individual checks.	Varies per check.	`RAT_{CHECK-ID}_TIMEOUT`
General timeout for all checks	90	`RAT_TIMEOUT`
SSH login DNS handshake. Specify the time in seconds for checking passwords on the remote nodes.	1	`RAT_PASSWORDCHECK_TIMEOUT`

   - The default timeouts are lengthy enough for most cases. If it is not long enough, then it is possible you are experiencing a system performance problem that should be corrected. Many timeouts can be indicative of a non-Oracle ORAchk and Oracle EXAchk problem in the environment.

3. If you can not increase the timeout, 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 users for parallel database run.

   - Database collections are run in parallel. The default number of slave processes used for parallel database run is calculated automatically. You can change the default number using the options:`-dbparallel` *`slave processes`*, or `-dbparallelmax`

     The higher the parallelism the more resources are consumed. However, the elapsed time is reduced. You can raise or lower the number of parallel slaves beyond the default value. After the entire system is brought up after maintenance, but before the users are permitted on the system, use a higher number of parallel slaves to finish a run as quickly as possible.

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

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

3 Proactively Detecting and Diagnosing Performance Issues for Oracle RAC

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

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

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

The ability of Oracle Cluster Health Advisor to detect performance and availability issues on Oracle Exadata systems has been improved in this release.

With the Oracle Cluster Health Advisor support for Oracle Solaris, you can now get early detection and prevention of performance and availability issues in your Oracle RAC database deployments.

For more information on *Installing Grid Infrastructure Management Repository*, see *Oracle® Grid Infrastructure Grid Infrastructure Installation and Upgrade Guide 20c for Linux*.

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

- Viewing the Status of Cluster Health Advisor
  SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.

Related Topics

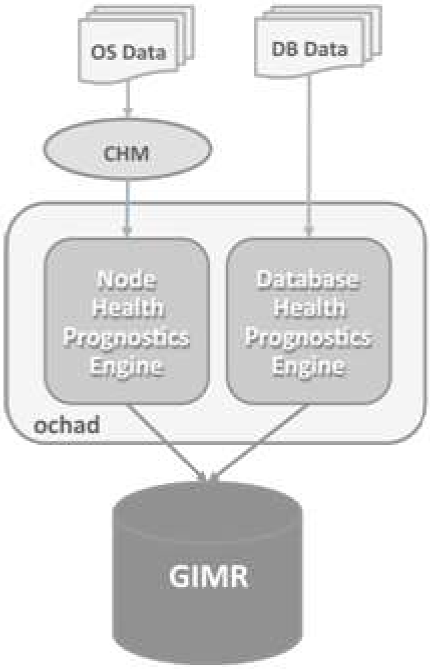
- Introduction to Oracle Cluster Health Advisor
  Oracle Cluster Health Advisor continuously monitors cluster nodes and Oracle RAC databases for performance and availability issue precursors to provide early warning of problems before they become critical.
- Installing Grid Infrastructure Management Repository

3.1 Oracle Cluster Health Advisor Architecture

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

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

Figure 3-1 Oracle Cluster Health Advisor Architecture


OS Data
DB Data
CHM
Node Health Prognostics Engine
Database Health Prognostics Engine
ochad
GIMR

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

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

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

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

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

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

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

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

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

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

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

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

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

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

   ```
   $ chactl status
   ```

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

3.3 Using Cluster Health Advisor for Health Diagnosis

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

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

To query the diagnostic data:

1. To query currently open problems, run the following command:

   ```
   chactl query diagnosis -db db_unique_name -start time -end time
   ```

   In the syntax example, *db_unique_name* is the name of your database instance. You also specify the start time and end time for which you want to retrieve data. Specify date and time in the `YYYY-MM-DD HH24:MI:SS` format.

2. Use the `-htmlfile` *`file_name`* option to save the output in HTML format.

Example 3-1 Cluster Health Advisor Output Examples in Text and HTML Format

This example shows the default text output for the `chactl query diagnosis` command for a database named *`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.

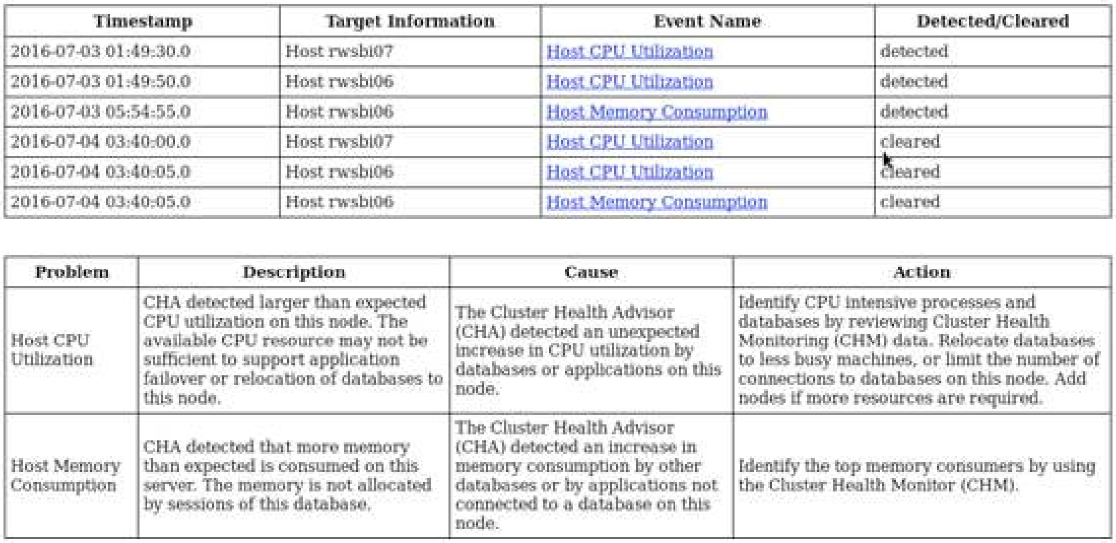
Note:

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 3-2 Cluster Health Advisor Diagnosis HTML Output



Timestamp	Target Information	Event Name	Detected/Cleared
2016-07-03 01:49:30.0	Host rwsbi07	Host CPU Utilization	detected
2016-07-03 01:49:50.0	Host rwsbi06	Host CPU Utilization	detected
2016-07-03 05:54:55.0	Host rwsbi06	Host Memory Consumption	detected
2016-07-04 03:40:00.0	Host rwsbi07	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host CPU Utilization	cleared
2016-07-04 03:40:05.0	Host rwsbi06	Host Memory Consumption	cleared

Problem	Description	Cause	Action
Host CPU Utilization	CHA detected larger than expected CPU utilization on this node. The available CPU resource may not be sufficient to support application failover or relocation of databases to this node.	The Cluster Health Advisor (CHA) detected an unexpected increase in CPU utilization by databases or applications on this node.	Identify CPU intensive processes and databases by reviewing Cluster Health Monitoring (CHM) data. Relocate databases to less busy machines, or limit the number of connections to databases on this node. Add nodes if more resources are required.
Host Memory Consumption	CHA detected that more memory than expected is consumed on this server. The memory is not allocated by sessions of this database.	The Cluster Health Advisor (CHA) detected an increase in memory consumption by other databases or by applications not connected to a database on this node.	Identify the top memory consumers by using the Cluster Health Monitor (CHM).

Related Topics

- chactl query diagnosis
  Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.

3.4 Calibrating an Oracle Cluster Health Advisor Model for a Cluster Deployment

As shipped with default node and database models, Oracle Cluster Health Advisor is designed not to generate false warning notifications.

You can increase the sensitivity and accuracy of the Oracle Cluster Health Advisor models for a specific workload using the `chactl calibrate` command.

Oracle recommends that a minimum of 6 hours of data be available and that both the cluster and databases use the same time range for calibration.

The `chactl calibrate` command analyzes a user-specified time interval that includes all workload phases operating normally. This data is collected while Oracle Cluster Health Advisor is monitoring the cluster and all the databases for which you want to calibrate.

1. To check if sufficient data is available, run the `query calibration` command.

   If 720 or more records are available, then Oracle Cluster Health Advisor successfully performs the calibration. The calibration function may not consider some data records to be normally occurring for the workload profile being used. In this case, filter the data by using the `KPISET` parameters in both the `query calibration` command and the `calibrate` command.

   For example:

   ```
   $ chactl query calibration -db oltpacdb -timeranges
   'start=2016-07-26 01:00:00,end=2016-07-26 02:00:00,start=2016-07-26
   03:00:00,end=2016-07-26 04:00:00'
   -kpiset 'name=CPUPERCENT min=20 max=40, name=IOTHROUGHPUT min=500
   max=9000' -interval 2
   ```

2. Start the calibration and store the model under a user-specified name for the specified date and time range.

   For example:

   ```
   $ chactl calibrate cluster –model weekday –timeranges ‘start=2016-07-03
   20:50:00,end=2016-07-04 15:00:00’
   ```

   After completing the calibration, Oracle Cluster Health Advisor automatically stores the new model in GIMR.

3. Use the new model to monitor the cluster as follows:

   For example:

   ```
   $ chactl monitor cluster –model weekday
   ```

Example 3-2   Output for the chactl query calibrate command

```
Database name : oltpacdb
Start time : 2016-07-26 01:03:10
End time : 2016-07-26 01:57:25
Total Samples : 120
Percentage of filtered data : 8.32%
The number of data samples may not be sufficient for calibration.

1) Disk read (ASM) (Mbyte/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
4.96      0.20      8.98      0.06      25.68

<25       <50       <75       <100      >=100
97.50%    2.50%     0.00%     0.00%     0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
27.73     9.72      31.75     4.16      109.39

<50       <100      <150      <200      >=200
73.33%    22.50%    4.17%     0.00%     0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
2407.50   1500.00   1978.55   700.00    7800.00

<5000     <10000    <15000    <20000    >=20000
83.33%    16.67%    0.00%     0.00%     0.00%

4) CPU utilization (total) (%)

MEAN      MEDIAN    STDDEV    MIN       MAX
21.99     21.75     1.36      20.00     26.80

<20       <40       <60       <80       >=80
0.00%     100.00%   0.00%     0.00%     0.00%

5) Database time per user call (usec/call)

MEAN      MEDIAN    STDDEV    MIN       MAX
267.39    264.87    32.05     205.80    484.57

<10000000  <20000000  <30000000  <40000000  <50000000  <60000000  <70000000
>=70000000
100.00%   0.00%     0.00%     0.00%     0.00%     0.00%     0.00%     0.00%

Database name : oltpacdb
Start time : 2016-07-26 03:00:00
End time : 2016-07-26 03:53:30
Total Samples : 342
Percentage of filtered data : 23.72%
The number of data samples may not be sufficient for calibration.
```

```
1) Disk read (ASM) (Mbyte/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
12.18     0.28      16.07     0.05      60.98

<25       <50       <75       <100      >=100
64.33%    34.50%    1.17%     0.00%     0.00%

2) Disk write (ASM) (Mbyte/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
57.57     51.14     34.12     16.10     135.29

<50       <100      <150      <200      >=200
49.12%    38.30%    12.57%    0.00%     0.00%

3) Disk throughput (ASM) (IO/sec)

MEAN      MEDIAN    STDDEV    MIN       MAX
5048.83   4300.00   1730.17   2700.00   9000.00

<5000     <10000    <15000    <20000    >=20000
63.74%    36.26%    0.00%     0.00%     0.00%

4) CPU utilization (total) (%)

MEAN      MEDIAN    STDDEV    MIN       MAX
23.10     22.80     1.88      20.00     31.40

<20       <40       <60       <80       >=80
0.00%     100.00%   0.00%     0.00%     0.00%

5) Database time per user call (usec/call)

MEAN      MEDIAN    STDDEV    MIN       MAX
744.39    256.47    2892.71   211.45    45438.35

<10000000  <20000000  <30000000  <40000000  <50000000  <60000000  <70000000
>=70000000
100.00%   0.00%     0.00%     0.00%     0.00%     0.00%     0.00%     0.00%
```

Related Topics

- chactl calibrate
  Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- chactl query calibration
  Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- chactl Command Reference
  The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.

3.5 Viewing the Details for an Oracle Cluster Health Advisor Model

Use the `chactl query model` command to view the model details.

- You can review the details of an Oracle Cluster Health Advisor model at any time using the `chactl query model` command.

  For example:

  ```
  $ chactl query model -name weekday
  Model: weekday
  Target Type: CLUSTERWARE
  Version: OS12.2_V14_0.9.8
  OS Calibrated on: Linux amd64
  Calibration Target Name: MYCLUSTER
  Calibration Date: 2016-07-05 01:13:49
  Calibration Time Ranges: start=2016-07-03 20:50:00,end=2016-07-04 15:00:00
  Calibration KPIs: not specified
  ```

  You can also rename, import, export, and delete the models.

3.6 Managing the Oracle Cluster Health Advisor Repository

Oracle Cluster Health Advisor repository stores the historical records of cluster host problems, database problems, and associated metric evidence, along with models.

The Oracle Cluster Health Advisor repository is used to diagnose and triage periodic problems. By default, the repository is sized to retain data for 16 targets (nodes and database instances) for 72 hours. If the number of targets increase, then the retention time is automatically decreased. Oracle Cluster Health Advisor generates warning messages when the retention time goes below 72 hours, and stops monitoring and generates a critical alert when the retention time goes below 24 hours.

Use CHACTL commands to manage the repository and set the maximum retention time.

1. To retrieve the repository details, use the following command:

   ```
   $ chactl query repository
   ```

   For example, running the command mentioned earlier shows the following output:

   ```
   specified max retention time(hrs) : 72
   available retention time(hrs)     : 212
   available number of entities      : 2
   allocated number of entities      : 0
   total repository size(gb)         : 2.00
   allocated repository size(gb)     : 0.07
   ```

2. To set the maximum retention time in hours, based on the current number of targets being monitored, use the following command:

   ```
   $ chactl set maxretention -time number_of_hours
   ```

   For example:

   ```
   $ chactl set maxretention -time 80
   max retention successfully set to 80 hours
   ```

   > **Note:**
   >
   > The `maxretention` setting limits the oldest data retained in the repository, but is not guaranteed to be maintained if the number of monitored targets increase. In this case, if the combination of monitored targets and number of hours are not sufficient, then increase the size of the Oracle Cluster Health Advisor repository.

3. To increase the size of the Oracle Cluster Health Advisor repository, use the `chactl resize repository` command.

   For example, to resize the repository to support 32 targets using the currently set maximum retention time, you would use the following command:

   ```
   $ chactl resize repository –entities 32
   repository successfully resized for 32 targets
   ```

3.7 Viewing the Status of Cluster Health Advisor

SRVCTL commands are the tools that offer total control on managing the life cycle of Oracle Cluster Health Advisor as a highly available service.

Use SRVCTL commands to 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.
   ```

Part II

Automatically Monitoring the Cluster

You can use components of Autonomous Health Framework to monitor your cluster on a regular basis.

- Collecting Operating System Resources Metrics
  CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.
- Monitoring System Metrics for Cluster Nodes
  This chapter explains the methods to monitor Oracle Clusterware.

4

Collecting Operating System Resources Metrics

CHM is a high-performance, lightweight daemon that collects, analyzes, aggregates, and stores a large set of operating system metrics to help you diagnose and troubleshoot system issues.

You can now configure Oracle Cluster Health Monitor to operate in local mode to report the operating system metrics using the `oclumon dumpnodeview local` command even if you have not deployed GIMR.

In local mode, you can get only the local node data. In earlier releases, Oracle Cluster Health Monitor required GIMR to report the operating system metrics using the `oclumon dumpnodeview` command.

Supported Platforms

Linux, Microsoft Windows, Solaris, AIX, IBM Z Series, and ARM

Why CHM is unique

CHM	Typical OS Collector
Last man standing - daemon runs memory locked, RT scheduling class ensuring consistent data collection under system load.	Inconsistent data dropouts due to scheduling delays under system load.
High fidelity data sampling rate, 5 seconds. Very low resource usage profile at 5-second sampling rates.	Executing multiple utilities creates additional overhead on the system being monitored, and worsens with higher sampling rates.
High Availability daemon, collated data collections across multiple resource categories. Highly optimized collector (data read directly from the operating system, same source as utilities).	Set of scripts/command-line utilities, for example, `top`, `ps`, `vmstat`, `iostat`, and so on re-directing their output to one or more files for every collection sample.
Collected data is collated into a system snapshot overview (**Nodeview**) on every sample, Nodeview also contains additional summarization and analysis of the collected data across multiple resource categories.	System snapshot overviews across different resource categories are very tedious to collate.
Significant inline analysis and summarization during data collection and collation into the Nodeview greatly reduces tedious, manual, time-consuming analysis to drive meaningful insights.	The analysis is time-consuming and processing-intensive as the output of various utilities across multiple files needs to be collated, parsed, interpreted, and then analyzed for meaningful insights.
Performs Clusterware-aware specific metrics collection (Process Aggregates, ASM/OCR/VD disk tagging, Private/Public NIC tagging). Also provides an extensive toolset for in-depth data analysis and visualization.	None

- Understanding Cluster Health Monitor Services
  Cluster Health Monitor uses system monitor (`osysmond`) and cluster logger (`ologgerd`) services to collect diagnostic data.
- Collecting Cluster Health Monitor Data
  Collect Cluster Health Monitor data from any node in the cluster.
- Operating System Metrics Collected by Cluster Health Monitor
  Review the metrics collected by CHM.
- Using Cluster Health Monitor from Enterprise Manager Cloud Control
  Histograms presented in real-time and historical modes enable you to understand precisely what was happening at the time of degradation or failure.

Related Topics

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

4.1 Understanding Cluster Health Monitor Services

Cluster Health Monitor uses system monitor (`osysmond`) and cluster logger (`ologgerd`) services to collect diagnostic data.

About the System Monitor Service

The system monitor service (`osysmond`) is a real-time monitoring and operating system metric collection service that runs on each cluster node. The system monitor service is managed as a High Availability Services (HAS) resource. The system monitor service forwards the collected metrics to the cluster logger service, `ologgerd`. The cluster logger service stores the data in the Oracle Grid Infrastructure Management Repository database.

In addition, `osysmond` persists the collected operating system metrics under a directory in ORACLE_BASE.

Metric Repository is auto-managed on the local filesystem. You can change the location and size of the repository.

- Nodeview samples are continuously written to the repository (JSON record)
- Historical data is auto-archived into hourly zip files
- Archived files are automatically purged once the default retention limit is reached (default: 200 MB)

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

Support for Deploying Grid Infrastructure Management Repository (GIMR) into a Separate Oracle Home

Starting with Oracle Grid Infrastructure 20c, you must configure the Grid Infrastructure Management Repository (GIMR) in a separate Oracle home, instead of in the Grid home. This option is available when you configure GIMR during a fresh Oracle Grid Infrastructure installation or you add a GIMR to an existing deployment. It is mandatory to configure GIMR in a separate Oracle home when you upgrade Oracle Grid infrastructure with an existing GIMR deployed in it.

A separate Oracle home for the GIMR ensures faster rolling upgrades, less errors, and fewer rollback situations. The Oracle Grid Infrastructure installation owner user must own the GIMR home.

For more information, see Installing Grid Infrastructure Management Repository

Remote GIMR Support for Oracle Standalone Clusters

The remote Grid Infrastructure Management Repository (GIMR) feature for Oracle Standalone Cluster enables you to use a centralized GIMR. This feature does not require local cluster resources to host the GIMR.

The remote GIMR feature provides access to a persistent data store that significantly enhances the proactive diagnostic functionality of Cluster Health Monitor, Cluster Health Advisor, and Autonomous Health Framework clients. The remote GIMR feature saves cost by freeing up local resources and licensed database server resources.

For more information, see Creating GIMR Credentials File for Oracle Standalone Clusters With Remote GIMR

4.2 Collecting Cluster Health Monitor Data

Collect Cluster Health Monitor data from any node in the cluster.

Oracle recommends that you run the `tfactl diagcollect` command to collect diagnostic data when an Oracle Clusterware error occurs.

4.3 Operating System Metrics Collected by Cluster Health Monitor

Review the metrics collected by CHM.

Overview of Metrics

CHM groups the operating system data collected into a **Nodeview**. A **Nodeview** is a grouping of metric sets where each metric set contains detailed metrics of a unique system resource.

Brief description of metric sets are as follows:

- **CPU metric set:** Metrics for top 127 CPUs sorted by usage percentage
- **Device metric set:** Metrics for 127 devices that include ASM/VD/OCR along with those having a high average wait time
- **Process metric set:** Metrics for 127 processes
  - Top 25 CPU consumers (idle processes not reported)

  - Top 25 Memory consumers (RSS < 1% of total RAM not reported)
  - Top 25 I/O consumers
  - Top 25 File Descriptors consumers (helps to identify top inode consumers)
  - Process Aggregation: Metrics summarized by foreground and background processes for all Oracle Database and Oracle ASM instances
- **Network metric set:** Metrics for 16 NICS that include public and private interconnects
- **NFS metric set:** Metrics for 32 NFS ordered by round trip time
- **Protocol metric set:** Metrics for protocol groups TCP, UDP, and IP
- **Filesystem metric set:** Metrics for filesystem utilization
- **Critical resources metric set:** Metrics for critical system resource utilization
  - CPU Metrics: system-wide CPU utilization statistics
  - Memory Metrics: system-wide memory statistics
  - Device Metrics: system-wide device statistics distinct from individual device metric set
  - NFS Metrics: Total NFS devices collected every 30 seconds
  - Process Metrics: system-wide unique process metrics

CPU Metric Set

Contains metrics from all CPU cores ordered by usage percentage.

Table 4-1 CPU Metric Set

Metric Name (units)	Description
system [%]	Percentage of CPU utilization occurred while executing at the system level (kernel).
user [%]	Percentage of CPU utilization occurred while executing at the user level (application).
usage [%]	Total utilization (**system[%]** + **user[%]**).
nice [%]	Percentage of CPU utilization occurred while executing at the user level with nice priority.
ioWait [%]	Percentage of time that the CPU was idle during which the system had an outstanding disk I/O request.
steal [%]	Percentage of time spent in involuntary wait by the virtual CPU while the hypervisor was servicing another virtual processor.

Device Metric Set

Contains metrics from all disk devices/partitions ordered by their service time in milliseconds.

Table 4-2 Device Metric Set

Metric Name (units)	Description
ioR [KB/s]	Amount of data read from the device.
ioW [KB/s]	Amount of data written to the device.
numIOs [#/s]	Average disk I/O operations.

Table 4-2 (Cont.) Device Metric Set

Metric Name (units)	Description
qLen [#]	Number of I/O queued requests, that is, in a wait state.
aWait [msec]	Average wait time per I/O.
svcTm [msec]	Average service time per I/O request.
util [%]	Percent utilization of the device (same as `'%util` metric from the `iostat -x` command. Represents the percentage of time device was active).

Process Metric Set

Contains multiple categories of summarized metric data computed across all system processes.

Table 4-3 Process Metric Set

Metric Name (units)	Description
pid	Process ID.
pri	Process priority (raw value from the operating system).
psr	The processor that process is currently assigned to or running on.
pPid	Parent process ID.
nice	Nice value of the process.
state	State of the process. For example, `R->Running`, `S->Interruptible sleep`, and so on.
class	Scheduling class of the process. For example, `RR->RobinRound`, `FF->First in First out`, `B->Batch scheduling`, and so on.
fd [#]	Number of file descriptors opened by this process, which is updated every 30 seconds.
name	Name of the process.
cpu [%]	Process CPU utilization across cores. For example, 50% => 50% of single core, 400% => 100% usage of 4 cores.
thrds [#]	Number of threads created by this process.
vmem [KB]	Process virtual memory usage (KB).
shMem [KB]	Process shared memory usage (KB).
rss [KB]	Process memory-resident set size (KB).
ioR [KB/s]	I/O read in kilobytes per second.
ioW [KB/s]	I/O write in kilobytes per second.
ioT [KB/s]	I/O total in kilobytes per second.
cswch [#/s]	Context switch per second. Collected only for a few critical Oracle Database processes.
nvcswch [#/s]	Non-voluntary context switch per second. Collected only for a few critical Oracle Database processes.
cumulativeCpu [ms]	Amount of CPU used so far by the process in microseconds.

NIC Metric Set

Contains metrics from all network interfaces ordered by their total rate in kilobytes per second.

Table 4-4 NIC Metric Set

Metric Name (units)	Description
name	Name of the interface.
tag	Tag for the interface, for example, **public**, **private**, and so on.
mtu [B]	Size of the maximum transmission unit in bytes supported for the interface.
rx [Kbps]	Average network receive rate.
tx [Kbps]	Average network send rate.
total [Kbps]	Average network transmission rate (**rx[Kb/s]** + **tx[Kb/s]**).
rxPkt [#/s]	Average incoming packet rate.
txPkt [#/s]	Average outgoing packet rate.
pkt [#/s]	Average rate of packet transmission (**rxPkt[#/s]** + **txPkt[#/s]**).
rxDscrd [#/s]	Average rate of dropped/discarded incoming packets.
txDscrd [#/s]	Average rate of dropped/discarded outgoing packets.
rxUnicast [#/s]	Average rate of unicast packets received.
rxNonUnicast [#/s]	Average rate of multicast packets received.
dscrd [#/s]	Average rate of total discarded packets (**rxDscrd** + **txDscrd**).
rxErr [#/s]	Average error rate for incoming packets.
txErr [#/s]	Average error rate for outgoing packets.
Err [#/s]	Average error rate of total transmission (**rxErr[#/s]** + **txErr[#/s]**).

NFS Metric Set

Contains top 32 NFS ordered by round trip time. This metric set is collected once every 30 seconds.

Table 4-5 NFS Metric Set

Metric Name (units)	Description
op [#/s]	Number of read/write operations issued to a filesystem per second.
bytes [#/sec]	Number of bytes read/write per second from a filesystem.
rtt [s]	This is the duration from the time that the client's kernel sends the RPC request until the time it receives the reply.
exe [s]	This is the duration from that NFS client does the RPC request to its kernel until the RPC request is completed, this includes the RTT time above.

Table 4-5 (Cont.) NFS Metric Set

Metric Name (units)	Description
retrains [%]	This is the retransmission's frequency in percentage.

Protocol Metric Set

Contains specific metrics for protocol groups TCP, UDP, and IP. Metric values are cumulative since the system starts.

Table 4-6 TCP Metric Set

Metric Name (units)	Description
failedConnErr [#]	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.
estResetErr [#]	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.
segRetransErr [#]	Total number of TCP segments retransmitted.
rxSeg [#]	Total number of TCP segments received on TCP layer.
txSeg [#]	Total number of TCP segments sent from TCP layer.

Table 4-7 UDP Metric Set

Metric Name (units)	Description
unkPortErr [#]	Total number of received datagrams for which there was no application at the destination port.
rxErr [#]	Number of received datagrams that could not be delivered for reasons other than the lack of an application at the destination port.
rxPkt [#]	Total number of packets received.
txPkt [#]	Total number of packets sent.

Table 4-8 IP Metric Set

Metric Name (units)	Description
ipHdrErr [#]	Number of input datagrams discarded due to errors in their IPv4 headers.
addrErr [#]	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.

Table 4-8 (Cont.) IP Metric Set

Metric Name (units)	Description
unkProtoErr [#]	Number of locally-addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.
reasFailErr [#]	Number of failures detected by the IPv4 reassembly algorithm.
fragFailErr [#]	Number of IPv4 discarded datagrams due to fragmentation failures.
rxPkt [#]	Total number of packets received on IP layer.
txPkt [#]	Total number of packets sent from IP layer.

Filesystem Metric Set

Contains metrics for filesystem utilization. Collected only for **GRID_HOME** filesystem.

Table 4-9 Filesystem Metric Set

Metric Name (units)	Description
mount	Mount point.
type	Filesystem type, for example, **etx4**.
tag	Filsystem tag, for example, **GRID_HOME**.
total [KB]	Total amount of space (KB).
used [KB]	Amount of used space (KB).
avbl [KB]	Amount of available space (KB).
used [%]	Percentage of used space.
ifree [%]	Percentage of free file nodes.

System Metric Set

Contains a summarized metric set of critical system resource utilization.

Table 4-10 CPU Metrics

Metric Name (units)	Description
pCpus [#]	Number of physical processing units in the system.
Cores [#]	Number of cores for all CPUs in the system.
vCpus [#]	Number of logical processing units in the system.
cpuHt	CPU Hyperthreading enabled (**Y**) or disabled (**N**).
osName	Name of the operating system.
chipName	Name of the chip of the processing unit.
system [%]	Percentage of CPUs utilization that occurred while executing at the system level (kernel).
user [%]	Percentage of CPUs utilization that occurred while executing at the user level (application).
usage [%]	Total CPU utilization (**system[%]** + **user[%]**).
nice [%]	Percentage of CPUs utilization occurred while executing at the user level with **NICE** priority.

Table 4-10 (Cont.) CPU Metrics

Metric Name (units)	Description
ioWait [%]	Percentage of time that the CPUs were idle during which the system had an outstanding disk I/O request.
Steal [%]	Percentage of time spent in involuntary wait by the virtual CPUs while the hypervisor was servicing another virtual processor.
cpuQ [#]	Number of processes waiting in the run queue within the current sample interval.
loadAvg1	Average system load calculated over time of one minute.
loadAvg5	Average system load calculated over of time of five minutes.
loadAvg15	Average system load calculated over of time of 15 minutes. High load averages imply that a system is overloaded; many processes are waiting for CPU time.
Intr [#/s]	Number of interrupts occurred per second in the system.
ctxSwitch [#/s]	Number of context switches that occurred per second in the system.

Table 4-11 Memory Metrics

Metric Name (units)	Description
totalMem [KB]	Amount of total usable RAM (KB).
freeMem [KB]	Amount of free RAM (KB).
avblMem [KB]	Amount of memory available to start a new process without swapping.
shMem [KB]	Memory used (mostly) by **tmpfs**.
swapTotal [KB]	Total amount of physical swap memory (KB).
swapFree [KB]	Amount of swap memory free (KB).
swpIn [KB/s]	Average swap in rate within the current sample interval (KB/sec).
swpOut [KB/s]	Average swap-out rate within the current sample interval (KB/sec).
pgIn [#/s]	Average page in rate within the current sample interval (pages/sec).
pgOut [#/s]	Average page out rate within the current sample interval (pages/sec).
slabReclaim [KB]	The part of the slab that might be reclaimed such as caches.
buffer [KB]	Memory used by kernel buffers.
Cache [KB]	Memory used by the page cache and slabs.
bufferAndCache [KB]	Total size of buffer and cache (**buffer[KB]** + **Cache[KB]**).
hugePageTotal [#]	Total number of huge pages present in the system for the current sample interval.

Table 4-11 (Cont.) Memory Metrics

Metric Name (units)	Description
hugePageFree [KB]	Total number of free huge pages in the system for the current sample interval.
hugePageSize [KB]	Size of one huge page in KB, depends on the operating system version. Typically the same for all samples for a particular host.

Table 4-12 Device Metrics

Metric Name (units)	Description
disks [#]	Number of disks configured in the system.
ioR [KB/s]	Aggregate read rate across all devices.
ioW [KB/s]	Aggregate write rate across all devices.
numIOs [#/s]	Aggregate I/O operation rate across all devices.

Table 4-13 NFS Metrics

Metric Name (units)	Description
nfs [#]	Total NFS devices.

Table 4-14 Process Metrics

Metric Name (units)	Description
fds [#]	Number of open file structs in system.
procs [#]	Number of processes.
rtProcs [#]	Number of real-time processes.
procsInDState	Number of processes in uninterruptible sleep.
sysFdLimit [#]	System limit on a number of file structs.
procsOnCpu [#]	Number of processes currently running on CPU.
procsBlocked [#]	Number of processes waiting for some event/ resource becomes available, such as for the completion of an I/O operation.

Process Aggregates Metric Set

Contains aggregated metrics for all processes by process groups.

Table 4-15 Process Aggregates Metric Set

Metric Name (units)	Description
DBBG	User Oracle Database background process group.
DBFG	User Oracle Database foreground process group.
MDBBG	MGMTDB background processes group.
MDBFG	MGMTDB foreground processes group.
ASMBG	ASM background processes group.

Table 4-15 (Cont.) Process Aggregates Metric Set

Metric Name (units)	Description
ASMFG	ASM foreground processes group.
IOXBG	IOS background processes group.
IOXFG	IOS foreground processes group.
APXBG	APX background processes group.
APXFG	APX foreground processes group.
CLUST	Clusterware processes group.
OTHER	Default group.

For each group, the below metrics are aggregated to report a group summary.

Metric Name (units)	Description
processes [#]	Total number of processes in the group.
cpu [%]	Aggregated CPU utilization.
rss [KB]	Aggregated resident set size.
shMem [KB]	Aggregated shared memory usage.
thrds [#]	Aggregated thread count.
fds [#]	Aggregated open file-descriptor.
cpuWeight [%]	Contribution of the group in overall CPU utilization of the machine.

4.4 Using Cluster Health Monitor from Enterprise Manager Cloud Control

Histograms presented in real-time and historical modes enable you to understand precisely what was happening at the time of degradation or failure.

The metric data from Cluster Health Monitor is available in graphical display within Enterprise Manager Cloud Control. Complete cluster views of this data are accessible from the cluster target page. Selecting the **Cluster Health Monitoring** menu item from the **Cluster** menu presents a log-in screen prompting for the Cluster Health Monitor credentials. There is a fixed EMUSER and the password is user-specified. Once the credentials are saved, you then can view Cluster Health Monitor data for the last day in overview format for the entire cluster. Metric categories are CPU, Memory, and Network.

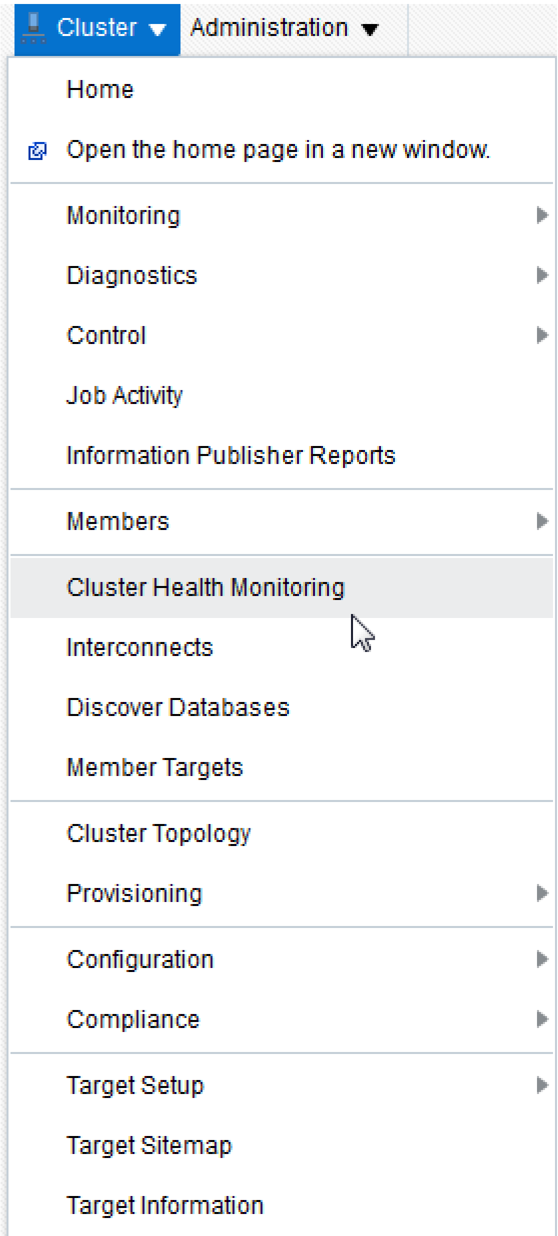
Each category is able to be separately display in greater detail showing more metrics. For example, selecting CPU results in cluster graphs detailing CPU System Usage, CPU User Usage, and CPU Queue Length. From any cluster view, you can select individual node views to more closely examine performance of a single server. As in the case of CPU, the performance of each core is displayed. Move your cursor along the graph to see a tool-tip displaying the numerical values and time stamp of that point.

Besides examining the performance of the current day, you can also review historical data. The amount of historical data is governed by the retention time configured in the Cluster Health Monitor repository in the 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 4-1 EMCC - Cluster Health Monitoring

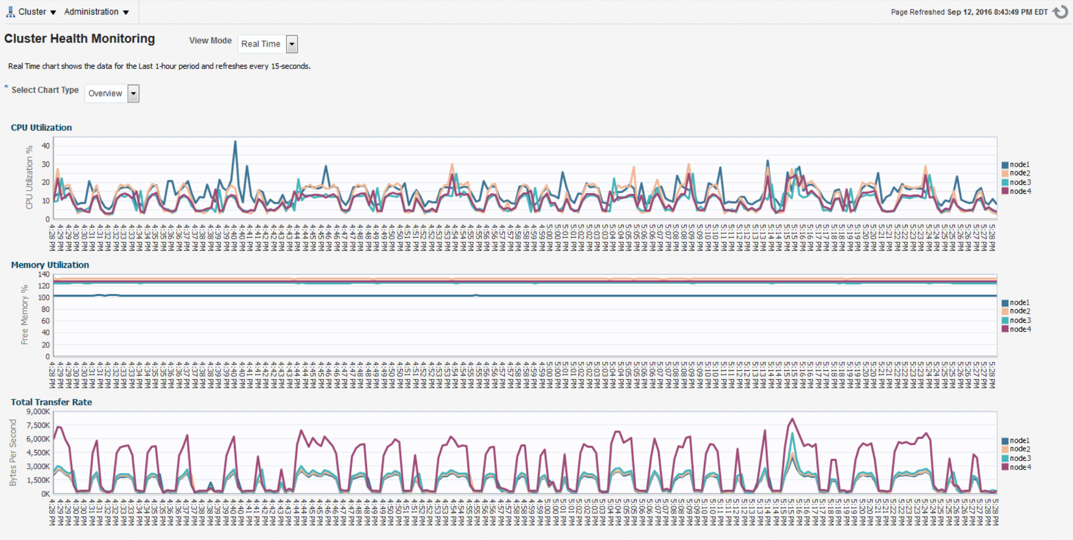


4. Enter Cluster Health Monitor login credentials.
5. From the **View Mode** drop-down list, select the **Real Time** option to view the current data.

By default, EMCC displays the **Overview** of resource utilization. You can filter by **CPU**, **Memory**, and **Network** by selecting an appropriate option from the **Select Chart Type** drop-down list.

While viewing CPU and Network metric graphs, click a node name on the legend to view more details.

Figure 4-2 Cluster Health Monitoring - Real Time Data



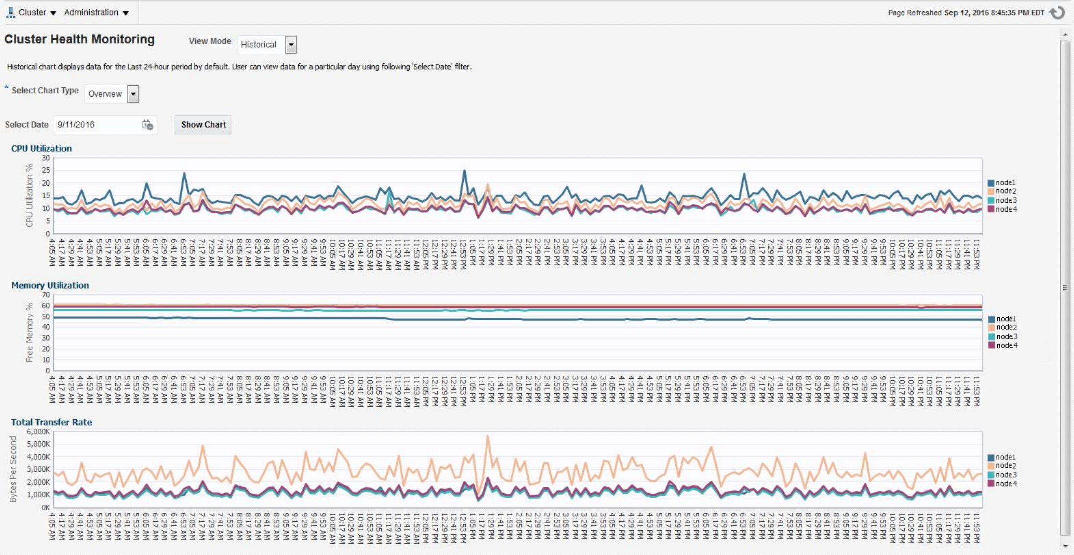
6. From the **View Mode** drop-down list, select the **Historical** option to view data for the last 24 hours.

   - To filter historical data by date, select a day on the **Select Date** calendar control and then click **Show Chart**.

   By default, EMCC displays the **Overview** of resource utilization. You can filter by **CPU**, **Memory**, and **Network** by selecting an appropriate option from the **Select Chart Type** drop-down list.

   While viewing CPU and Network metric graphs, click a node name on the legend to view more details.

Figure 4-3   Cluster Health Monitoring - Historical Data



Cluster Health Monitoring

View Mode: Historical

Historical chart displays data for the Last 24-hour period by default. User can view data for a particular day using following 'Select Date' filter.

* Select Chart Type: Overview

Select Date: 9/11/2016

5 Monitoring System Metrics for Cluster Nodes

This chapter explains the methods to monitor Oracle Clusterware.

Oracle recommends that you use Oracle Enterprise Manager to monitor everyday operations of Oracle Clusterware.

Cluster Health Monitor monitors the complete technology stack, including the operating system, ensuring smooth cluster operations. Both the components are enabled, by default, for any Oracle cluster. Oracle strongly recommends that you use both the components. Also, monitor Oracle Clusterware-managed resources using the Clusterware resource activity log.

- Monitoring Oracle Clusterware with Oracle Enterprise Manager
  Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.
- Monitoring Oracle Clusterware with Cluster Health Monitor
  You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.
- Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures
  The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

Related Topics

- Managing the Cluster Resource Activity Log
  Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.

5.1 Monitoring Oracle Clusterware with Oracle Enterprise Manager

Use Oracle Enterprise Manager to monitor the Oracle Clusterware environment.

When you log in to Oracle Enterprise Manager using a client browser, the **Cluster Database Home** page appears where you can monitor the status of both Oracle Database and Oracle Clusterware environments. Oracle Clusterware monitoring includes the following details:

- Current and historical Cluster Health Monitor data in Oracle Enterprise Manager on the cluster target
- Notifications if there are any VIP relocations
- Status of the Oracle Clusterware on each node of the cluster using information obtained through the Cluster Verification Utility (CVU)
- Notifications if node applications (`nodeapps`) start or stop
- Notification of issues in the Oracle Clusterware alert log for the Oracle Cluster Registry, voting file issues (if any), and node evictions

The **Cluster Database Home** page is similar to a single-instance Database Home page. However, on the Cluster Database Home page, Oracle Enterprise Manager displays the system state and availability. The system state and availability includes a summary about alert messages and job activity, and links to all the database and Oracle Automatic Storage Management (Oracle ASM) instances. For example, track problems with services on the cluster including when a service is not running on all the preferred instances or when a service response time threshold is not being met.

Use the Oracle Enterprise Manager **Interconnects** page to monitor the Oracle Clusterware environment. The Interconnects page displays the following details:

- Public and private interfaces on the cluster
- Overall throughput on the private interconnect
- Individual throughput on each of the network interfaces
- Error rates (if any)
- Load contributed by database instances on the interconnect
- Notifications if a database instance is using public interface due to misconfiguration
- Throughput contributed by individual instances on the interconnect

All the information listed earlier is also available as collections that have a historic view. The historic view is useful with cluster cache coherency, such as when diagnosing problems related to cluster wait events. Access the Interconnects page by clicking the **Interconnect** tab on the Cluster Database home page.

Also, the Oracle Enterprise Manager **Cluster Database Performance** page provides a quick glimpse of the performance statistics for a database. Statistics are rolled up across all the instances in the cluster database in charts. Using the links next to the charts, you can get more specific information and perform any of the following tasks:

- Identify the causes of performance issues
- Decide whether resources must be added or redistributed
- Tune your SQL plan and schema for better optimization
- Resolve performance issues

The charts on the Cluster Database Performance page include the following:

- **Chart for Cluster Host Load Average**: The **Cluster Host Load Average** chart in the Cluster Database Performance page shows potential problems that are outside the database. The chart shows maximum, average, and minimum load values for available nodes in the cluster for the previous hour.
- **Chart for Global Cache Block Access Latency**: Each cluster database instance has its own buffer cache in its System Global Area (SGA). Using Cache Fusion, Oracle RAC environments logically combine buffer cache of each instance to enable the database instances to process data as if the data resided on a logically combined, single cache.
- **Chart for Average Active Sessions**: The **Average Active Sessions** chart in the Cluster Database Performance page shows potential problems inside the database. Categories, called wait classes, show how much of the database is using a resource, such as CPU or disk I/O. Comparing CPU time to wait time helps to determine how much of the response time is consumed with useful work rather than waiting for resources that are potentially held by other processes.
- **Chart for Database Throughput**: The **Database Throughput** charts summarize any resource contention that appears in the Average Active Sessions chart, and also show how much work the database is performing on behalf of the users or applications. The **Per Second** view shows the number of transactions compared to the number of logons, and the amount of physical reads compared to the redo size for each second. The **Per Transaction** view shows the amount of physical reads compared to the redo size for each transaction. Logons is the number of users that are logged on to the database.

In addition, the **Top Activity** drop-down menu on the **Cluster Database Performance** page enables you to see the activity by wait events, services, and instances. In addition, you can see the details about SQL/sessions by going to a prior point in time by moving the slider on the chart.

5.2 Monitoring Oracle Clusterware with Cluster Health Monitor

You can use the OCLUMON command-line tool to interact with Cluster Health Monitor.

OCLUMON is included with Cluster Health Monitor. You can use it to query the Cluster Health Monitor repository to display node-specific metrics for a specified time period. You can also use OCLUMON to perform miscellaneous administrative tasks, such as the following:

- Changing the debug levels with the `oclumon debug` command
- Querying the version of Cluster Health Monitor with the `oclumon version` command
- Viewing the collected information in the form of a node view using the `oclumon dumpnodeview` command
- Changing the metrics database size using the `oclumon manage` command

Related Topics

- OCLUMON Command Reference
  Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.

5.3 Using the Cluster Resource Activity Log to Monitor Cluster Resource Failures

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs.

If an Oracle Clusterware-managed resource fails, then Oracle Clusterware logs messages about the failure in the **cluster resource activity log** located in the Grid Infrastructure Management Repository. Failures can occur as a result of a problem with a resource, a hosting node, or the network. The cluster resource activity log provides a unified view of the cause of resource failure.

Writes to the cluster resource activity log are tagged with an activity ID and any related data gets the same parent activity ID, and is nested under the parent data. For example, if Oracle Clusterware is running and you run the `crsctl stop clusterware -all` command, then all activities get activity IDs, and related activities are tagged with the same parent activity ID. On each node, the command creates sub-IDs under the parent IDs, and tags each of the respective activities with their corresponding activity ID. Further, each resource on the individual nodes creates sub-IDs based on the parent ID, creating a hierarchy of activity IDs. The hierarchy of activity IDs enables you to analyze the data to find specific activities.

For example, you may have many resources with complicated dependencies among each other, and with a database service. On Friday, you see that all of the resources are running on one node but when you return on Monday, every resource is on a different node, and you want to know why. Using the `crsctl query calog` command, you can query the cluster resource activity log for all activities involving those resources and the database service. The output provides a complete flow and you can query each sub-ID within the parent service failover ID, and see, specifically, what happened and why.

You can query any number of fields in the cluster resource activity log using filters. For example, you can query all the activities written by specific operating system users such as `root`. The output produced by the `crsctl query calog` command can be displayed in either a tabular format or in XML format.

The cluster resource activity log is an adjunct to current Oracle Clusterware logging and alert log messages.

> **Note:**
>
> Oracle Clusterware does not write messages that contain security-related information, such as log-in credentials, to the cluster activity log.

Use the following commands to manage and view the contents of the cluster resource activity log:

Part III

Automatic Problem Solving

Some situations can be automatically resolved with tools in the Autonomous Health Framework.

- Resolving Database and Database Instance Delays
  Hang Manager preserves the database performance by resolving delays and keeping the resources available.

6

Resolving Database and Database Instance Delays

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

- Hang Manager Architecture
  Hang Manager autonomously runs as a `DIA0` task within the database.
- Optional Configuration for Hang Manager
  You can adjust the sensitivity, and control the size and number of the log files used by Hang Manager.
- Hang Manager Diagnostics and Logging
  Hang Manager autonomously resolves hangs and continuously logs the resolutions in the database alert logs and the diagnostics in the trace files.

Related Topics

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

6.1 Hang Manager Architecture

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

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 delay. If the sessions are suspected as delayed, Hang Manager then waits for a certain threshold time period to ensure that the sessions are delayed.
- **Verify:** In this phase, after the threshold time period is up, Hang Manager verifies that the sessions are delayed and selects a session that's causing the delay.

After selecting the session that's causing the delay, Hang Manager applies resolution methods on that session. If the chain of sessions or the delay resolves automatically, then Hang Manager does not apply delay resolution methods. However, if the delay does not resolve by itself, then Hang Manager resolves the delay by terminating the session that's causing the delay. 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.

For example, if a high rank session is included in the chain of delayed sessions, then Hang Manager expedites the termination of the session that's causing the delay. Termination of the session that's causing the delay prevents the high rank session from waiting too long and helps to maintain performance objective of the high rank session.

6.2 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 delay, then Hang Manager waits for a certain threshold time period to ensure that the sessions are delayed. 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 delays 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.

6.3 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 6-1   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 6-2   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 6-3 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.
```

Part IV

Collecting Diagnostic Data and Triaging, Diagnosing, and Resolving Issues

Use the tools available with Autonomous Health Framework to troubleshoot errors.

- Using Automatic Diagnostic Collections
  Oracle Trace File Analyzer monitors your logs for significant problems, such as internal errors like `ORA-00600`, or node evictions.
- Using On-Demand Diagnostic Collections
  Run Oracle Trace File Analyzer on demand using `tfactl` command-line tool.
- Using REST Service
  Learn to configure REST service, and use REST service APIs.
- Managing and Configuring Oracle Trace File Analyzer
  This section helps you manage Oracle Trace File Analyzer daemon, diagnostic collections, and the collection repository.
- Managing Oracle Database and Oracle Grid Infrastructure Logs
  This section enables you to manage Oracle Database and Oracle Grid Infrastructure diagnostic data and disk usage snapshots.

7 Using Automatic Diagnostic Collections

Oracle Trace File Analyzer monitors your logs for significant problems, such as internal errors like `ORA-00600`, or node evictions.

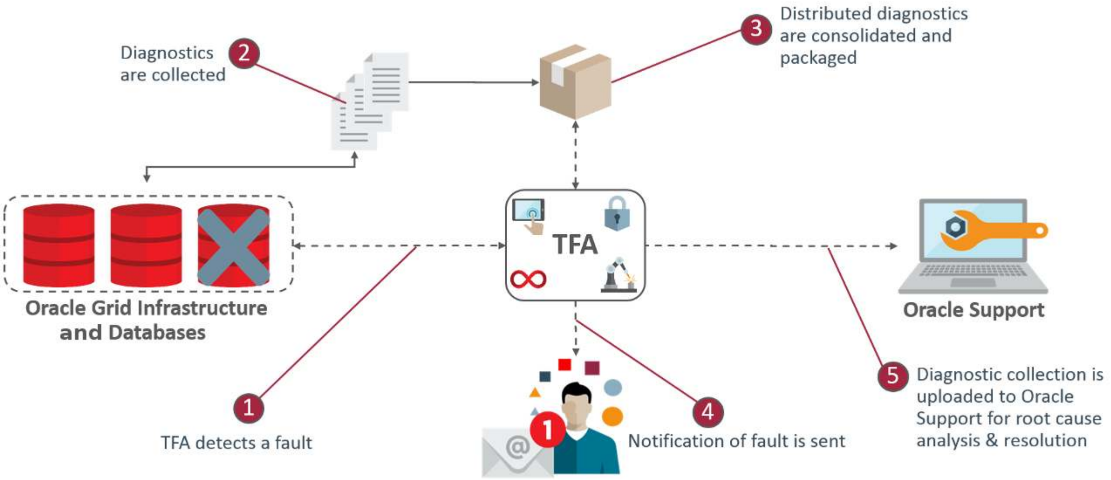
7.1 Collecting Diagnostics Automatically

This section explains automatic diagnostic collection concepts.

If Oracle Trace File Analyzer detects any problems, then it performs the following actions:

- Runs necessary diagnostics and collects all relevant log data at the time of a problem
- Trims log files to collect only what is necessary for diagnosis
- Collects and packages all trimmed diagnostics from all nodes in the cluster, consolidating everything on a single node
- Stores diagnostic collections in the Oracle Trace File Analyzer repository
- Sends you email notification of the problem and details of diagnostic collection that is ready for upload to Oracle Support

Figure 7-1 Automatic Diagnostic Collections


Diagnostics are collected
2
3
Distributed diagnostics are consolidated and packaged
TFA
Oracle Grid Infrastructure and Databases
Oracle Support
1
TFA detects a fault
1
4
Notification of fault is sent
5
Diagnostic collection is uploaded to Oracle Support for root cause analysis & resolution

Oracle Trace File Analyzer has a mechanism that prevents repeat errors from overwhelming your system with excessive, automatic collections.

Identifying an event triggers the start point for a collection and five minutes later Oracle Trace File Analyzer starts collecting diagnostic data. Starting five minutes later enables Oracle Trace File Analyzer to capture other relevant events in one operation. If events are still occurring after five minutes, then diagnostic collection continues to wait. Oracle Trace File Analyzer waits for 30 seconds with no events occurring up to an additional five minutes.

If events continue after 10 minutes, then Oracle Trace File Analyzer continues to perform diagnostic collection.

After completing the diagnostic collections, Oracle Trace File Analyzer sends email notifications that include the collection location to the designated recipients.

If your environment can make a connection to **oracle.com**, then you can use Oracle Trace File Analyzer to upload the collection to a Service Request.

```
$ tfactl set autodiagcollect=ON|OFF
```

Automatic collections are `ON` by default.

Table 7-1 Log Entries that Trigger Automatic collection

String Pattern	Log Monitored
`ORA-297(01|02|03|08|09|10|40)`	Alert Log - Oracle Database
`ORA-00600`	Alert Log - Oracle ASM
`ORA-07445`	Alert Log - Oracle ASM Proxy
`ORA-04(69|([7-8][0-9]|9([0-3]|[5-8])))`	Alert Log - Oracle ASM IO Server
`ORA-32701`	
`ORA-00494`	
`ORA-04020`	
`ORA-04021`	
`ORA-01578`	
`ORA-00700`	
System State dumped	
`CRS-016(07|10|11|12)`	Alert Log - Oracle Clusterware

Additionally, when Oracle Cluster Health Advisor detects a problem event, Oracle Trace File Analyzer automatically triggers the relevant diagnostic collection.

7.2 Configuring Email Notification Details

Configure Oracle Trace File Analyzer to send an email to the registered email address after an automatic collection completes.

To send emails, configure the system on which Oracle Trace Analyzer is running. You must configure notification with a user email address.

To configure email notification details:

1. To set the notification email for a specific `ORACLE_HOME`, include the operating system owner in the command:

   ```
   tfactl set notificationAddress=os_user:email
   ```

   For example:

   ```
   tfactl set notificationAddress=oracle:some.body@example.com
   ```

2. To set the notification email for any `ORACLE_HOME`:

   ```
   tfactl set notificationAddress=email
   ```

   For example:

   ```
   tfactl set notificationAddress=another.body@example.com
   ```

3. Configure the SMTP server using `tfactl set smtp`.

   Set the SMTP parameters when prompted.

Table 7-2 tfactl diagnosetfa Command Parameters

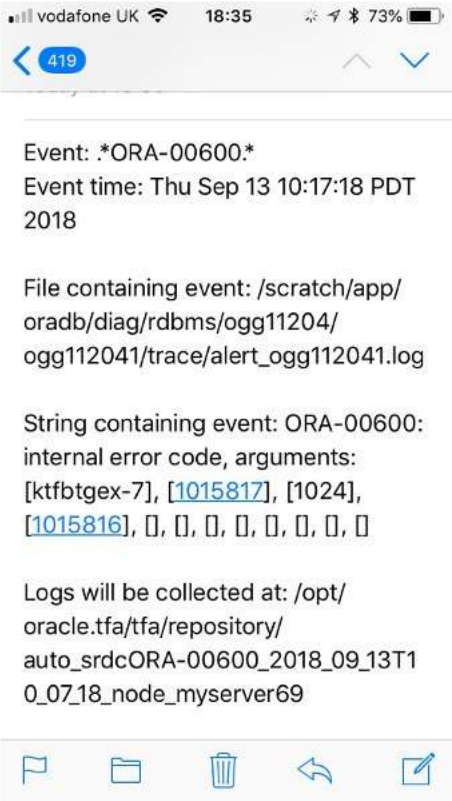
Parameter	Description
`smtp.host`	Specify the SMTP server host name.
`smtp.port`	Specify the SMTP server port.
`smtp.user`	Specify the SMTP user.
`smtp.password`	Specify password for the SMTP user.
`smtp.auth`	Set the Authentication flag to true or false.
`smtp.ssl`	Set the SSL flag to true or false.
`smtp.from`	Specify the from mail ID.
`smtp.to`	Specify the comma-delimited list of recipient mail IDs.
`smtp.cc`	Specify the comma-delimited list of CC mail IDs.
`smtp.bcc`	Specify the comma-delimited list of BCC mail IDs.
`smtp.debug`	Set the Debug flag to true or false.

> **Note:**
>
> You can view current SMTP configuration details using `tfactl print smtp`.

4. Verify SMTP configuration by sending a test email using `tfactl sendmail` *`email_address`*.

   If Oracle Trace File Analyzer detects that a significant error has occurred, then it sends an email notification as follows:

Figure 7-2 Email Notification



Event: .*ORA-00600.*
Event time: Thu Sep 13 10:17:18 PDT 2018

File containing event: /scratch/app/oradb/diag/rdbms/ogg11204/ogg112041/trace/alert_ogg112041.log

String containing event: ORA-00600: internal error code, arguments: [ktfbtgex-7], [1015817], [1024], [1015816], [], [], [], [], [], [], [], []

Logs will be collected at: /opt/oracle.tfa/tfa/repository/auto_srdcORA-00600_2018_09_13T10_07_18_node_myserver69

5. Do the following after receiving the notification email:
   a. To find the root cause, inspect the referenced collection details.
   b. If you can fix the issue, then resolve the underlying cause of the problem.
   c. If you do not know the root cause of the problem, then log an SR with Oracle Support, and upload the collection details.

7.3 Collecting Problems Detected by Oracle Cluster Health Advisor

Configure Oracle Cluster Health Advisor to automatically collect diagnostics for abnormal events, and send email notifications.

1. To configure Oracle Cluster Health Advisor auto collection for abnormal events:

```
tfactl set chaautocollect=ON
```

2. To enable Oracle Cluster Health Advisor notification through Oracle Trace File Analyzer:

```
tfactl set chanotification=on
```

3. To configure an email address for Oracle Cluster Health Advisor notifications to be sent to:

```
tfactl set notificationAddress=chatfa:john.doe@acompany.com
```

7.4 Sanitizing Sensitive Information in Oracle Trace File Analyzer Collections

After collecting copies of diagnostic data, Oracle Trace File Analyzer uses Adaptive Classification and Redaction (ACR) to sanitize sensitive data in the collections.

To mask or sanitize sensitive data in collections:

```
tfactl set redact=mask|sanitize|none
```

`mask`: blocks out the sensitive data in all collections, for example, replaces *myhost1* with *******

`sanitize`: replaces the sensitive data in all collections with random characters, for example, replaces myhost1 with *orzhmv1*

`none` (default): does not mask or sanitize sensitive data in collections

You can use the `-sanitize` and `-mask` options with the `diagcollect` command to sanitize or mask sensitive data in a specific collection.

To mask sensitive data:

1. To mask sensitive data in all collections:

   ```
   tfactl set redact=mask
   ```

2. To sanitize sensitive data in all collections:

   ```
   tfactl set redact=sanitize
   ```

3. To mask or sanitize sensitive data in a specific collection:

   For example:

   ```
   tfactl diagcollect -SRDC ORA-00600 -mask
   ```

   ```
   tfactl diagcollect -SRDC ORA-00600 -sanitize
   ```

7.5 Flood Control for Similar Issues

Flood control mechanism helps you save resource through fewer repeat collections for similar issues.

You can:

- Enable or disable flood control.
- How many times to collect for an event.
- Pause flood control.

The flood control data is stored in Berkeley Database and persists across Oracle Trace File Analyzer restarts.

Example 7-1 Flood Control Examples

To check if flood control is enabled or disabled:

```
# tfactl get floodcontrol
.----------------------------------------.
|               testhost                 |
+--------------------------------+-------+
| Configuration Parameter        | Value |
+--------------------------------+-------+
| Flood Control ( floodcontrol ) | ON    |
'--------------------------------+-------'
```

To check flood control limit:

```
# tfactl get fc.limit
.------------------------------------------------.
|                   testhost                     |
+----------------------------------------+-------+
| Configuration Parameter                | Value |
+----------------------------------------+-------+
| Flood Control Limit Count ( fc.limit ) | 3     |
'----------------------------------------+-------'
```

To check flood control limit time:

```
# tfactl get fc.limittime
.-------------------------------------------------------------.
|                          testhost                           |
+-----------------------------------------------------+-------+
| Configuration Parameter                             | Value |
+-----------------------------------------------------+-------+
| Flood Control Limit Time (minutes) ( fc.limitTime ) | 60    |
'-----------------------------------------------------+-------'
```

To check flood control pause time:

```
# tfactl get fc.pausetime
.-------------------------------------------------------------.
|                          testhost                           |
+-----------------------------------------------------+-------+
| Configuration Parameter                             | Value |
+-----------------------------------------------------+-------+
| Flood Control Pause Time (minutes) ( fc.pauseTime ) | 120   |
'-----------------------------------------------------+-------'
```

To print flood control details:

```
# tfactl floodcontrol print

.------------------------------------------------------------------------------
-----------------------------------------------------------------------------.
| Event                  | Count | Start Date                   | Last
Date                    | Limit | Limit Time | Pause Time | Coll Count | Skip
```

```
Count |
+------------------------+-------+------------------------------
+------------------------------+-------+------------+------------+------------
+------------+
| orcl:ORA-00600:user1   |     1 | Thu May 21 09:18:56 UTC 2020 | Thu May 21
09:18:56 UTC 2020 |     3 |         60 |        120 |          1 |          0
|
+------------------------+-------+------------------------------
+------------------------------+-------+------------+------------+------------
+------------+
| orcl:ORA-00600:user2   |     1 | Thu May 21 09:18:25 UTC 2020 | Thu May 21
09:18:25 UTC 2020 |     3 |         60 |        120 |          4 |          2
|
'------------------------+-------+------------------------------
+------------------------------+-------+------------+------------+------------
+------------'
```

To clear flood control:

```
# tfactl floodcontrol clear -event orcl:ORA-00600:user1
Successfully cleared Event orcl:ORA-00600:user1

# tfactl floodcontrol print
.-----------------------------------------------------------------------------
----------------------------------------.
| Event                  | Count | Start Date | Last Date | Limit | Limit
Time | Pause Time | Coll Count | Skip Count |
+------------------------+-------+------------+-----------+-------
+------------+------------+------------+------------+
| orcl:ORA-00600:user1   |     0 | null       | null      |     3 |
60 |        120 |          3 |          2 |
'------------------------+-------+------------+-----------+-------
+------------+------------+------------+------------'
```

To udate flood control details:

```
# tfactl floodcontrol update -event orcl:ORA-00600:user1 -limit 10 -limittime
90 -pausetime 180
Successfully updated Flood Control Event

# tfactl floodcontrol print -event orcl:ORA-00600:user1
.-----------------------------------------------------------------------------
-----------------------------------------------------------------------------.
| Event                  | Count | Start Date                   | Last
Date                    | Limit | Limit Time | Pause Time | Coll Count | Skip
Count |
+------------------------+-------+------------------------------
+------------------------------+-------+------------+------------+------------
+------------+
| orcl:ORA-00600:user1   |     1 | Thu May 21 09:18:25 UTC 2020 | Thu May 21
09:18:25 UTC 2020 |    10 |         90 |        180 |          4 |          2
|
'------------------------+-------+------------------------------
```

```
+------------------------------+-------+------------+------------+------------
+------------'
```

8 Using On-Demand Diagnostic Collections

Run Oracle Trace File Analyzer on demand using `tfactl` command-line tool.

- Collecting Diagnostics and Analyzing Logs On-Demand
  The `tfactl` command uses a combination of different Oracle Database support tools when it performs analysis.
- Viewing System and Cluster Summary
  The summary command gives you a real-time report of system and cluster status.
- Investigating Logs for Errors
  Use Oracle Trace File Analyzer to analyze all of your logs across your cluster to identify recent errors.
- Analyzing Logs Using the Oracle Database Support Tools
  The Oracle Database support tools bundle is available only when you download Oracle Trace File Analyzer from My Oracle Support note 2550798.1.
- Searching Oracle Trace File Analyzer Metadata
  You can search all metadata stored in the Oracle Trace File Analyzer index using `tfactl search -showdatatypes|-json [json_details]`.
- Oracle Trace File Analyzer Service Request Data Collections (SRDCs)
  Oracle Trace File Analyzer Service Request Data Collections (SRDCs) enable you to quickly collect the right diagnostic data.
- Diagnostic Upload
  Diagnostic upload eliminates the need for different set of commands to upload Oracle ORAchk, Oracle EXAchk, and Oracle Trace File Analyzer diagnostic collections to AHF Service, database, and Oracle Support.
- Changing Oracle Grid Infrastructure Trace Levels
  Enable trace levels to collect enough diagnostics to diagnose the cause of the problem.
- Performing Custom Collections
  Use the custom collection options to change the diagnostic collections from the default.

8.1 Collecting Diagnostics and Analyzing Logs On-Demand

The `tfactl` command uses a combination of different Oracle Database support tools when it performs analysis.

The `tfactl` command enables you to access Oracle Database support tools using common syntax. Using common syntax hides the complexity of the syntax differences between the tools.

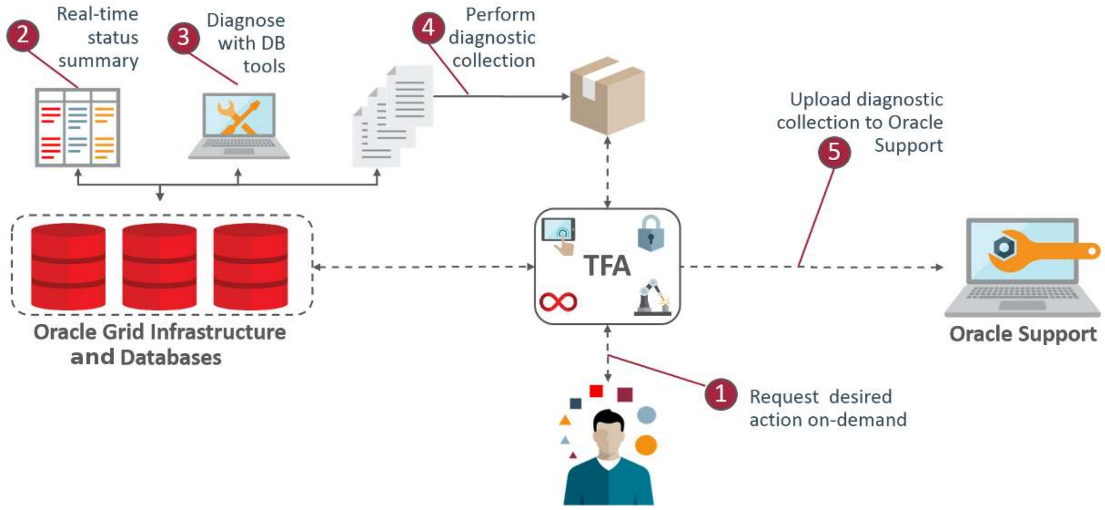
Use the Oracle Trace File Analyzer tools to perform analysis and resolve problems. If you need more help, then use the `tfactl` command to collect diagnostics for Oracle Support.

Oracle Trace File Analyzer does the following:

- Collects all relevant log data from a time of your choosing.
- Trims log files to collect only what is necessary for diagnosis.

- Packages all diagnostics on the node where `tfactl` was run from.

Figure 8-1 On-Demand Collections


2
Real-time status summary
3
Diagnose with DB tools
4
Perform diagnostic collection
Upload diagnostic collection to Oracle Support
5
Oracle Grid Infrastructure and Databases
TFA
Oracle Support
1
Request desired action on-demand

8.2 Viewing System and Cluster Summary

The summary command gives you a real-time report of system and cluster status.

Syntax

```
tfactl summary [options]
```

For more help use:

```
tfactl summary -help
```

8.3 Investigating Logs for Errors

Use Oracle Trace File Analyzer to analyze all of your logs across your cluster to identify recent errors.

1. To find all errors in the last one day:

   ```
   $ tfactl analyze -last 1d
   ```

2. To find all errors over a specified duration:

   ```
   $ tfactl analyze -last 18h
   ```

3. To find all occurrences of a specific error on any node, for example, to report `ORA-00600` errors:

   ```
   $ tfactl analyze -search “ora-00600" -last 8h
   ```

Related Topics

- tfactl summary
  Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.
- tfactl analyze
  Use the `tfactl analyze` command to obtain analysis of your system by parsing the database, Oracle Automatic Storage Management (Oracle ASM), and Oracle Grid Infrastructure alert logs, system message logs, OSWatcher Top, and OSWatcher Slabinfo files.

8.4 Analyzing Logs Using the Oracle Database Support Tools

The Oracle Database support tools bundle is available only when you download Oracle Trace File Analyzer from My Oracle Support note 2550798.1.

Oracle Trace File Analyzer with Oracle Database support tools bundle includes the following tools:

Table 8-1 Tools Included in Linux and UNIX

Tool	Description
`orachk` or `exachk`	Provides health checks for the Oracle stack. Oracle Autonomous Health Framework installs either Oracle EXAchk for engineered systems or Oracle ORAchk for all non-engineered systems. For more information, see My Oracle Support notes 1070954.1 and 2550798.1.
`oswatcher (oswbb)`	Collects and archives operating system metrics. These metrics are useful for instance or node evictions and performance Issues. For more information, see My Oracle Support note 301137.1.
`procwatcher (prw)`	Automates and captures database performance diagnostics and session level hang information. For more information, see My Oracle Support note 459694.1.
`oratop`	Provides near real-time database monitoring. For more information, see My Oracle Support note 1500864.1.
`alertsummary`	Provides summary of events for one or more database or Oracle ASM alert files from all nodes.
`ls`	Lists all files that Oracle Trace File Analyzer knows about for a given file name pattern across all nodes.
`pstack`	Generates the process stack for the specified processes across all nodes.
`grep`	Searches for a given string in the alert or trace files with a specified database.
`summary`	Provides high-level summary of the configuration.
`vi`	Opens alert or trace files for viewing a given database and file name pattern in the `vi` editor.
`tail`	Runs a tail on an alert or trace files for a given database and file name pattern.
`param`	Shows all database and operating system parameters that match a specified pattern.
`dbglevel`	Sets and unsets multiple Oracle Clusterware trace levels with one command.
`history`	Shows the shell history for the `tfactl` shell.

Table 8-1 (Cont.) Tools Included in Linux and UNIX

Tool	Description
`changes`	Reports changes in the system setup over a given time period. The report includes database parameters, operating system parameters, and the patches that are applied.
`calog`	Reports major events from the cluster event log.
`events`	Reports warnings and errors in the logs.
`managelogs`	Shows disk space usage and purges Automatic Diagnostic Repository (ADR) log and trace files.
`ps`	Finds processes.
`triage`	Summarizes `oswatcher` or `exawatcher` data.

Table 8-2 Tools Included in Microsoft Windows

Tool	Description
`calog`	Reports major events from the cluster event log.
`changes`	Reports changes in the system setup over a given time period. The report includes database parameters, operating system parameters, and patches applied.
`dir`	Lists all files Oracle Trace File Analyzer knows about for a given file name pattern across all nodes.
`events`	Reports warnings and errors seen in the logs.
`findstr`	Searches for a given string in the alert or trace files with a specified database.
`history`	Shows the shell history for the `tfactl` shell.
`managelogs`	Shows disk space usage and purges ADR log and trace files.
`notepad`	Opens alert or trace files for viewing a given database and file name pattern in the `notepad` editor.
`param`	Shows all database and operating system parameters that match a specified pattern.
`summary`	Provides high-level summary of the configuration.
`tasklist`	Finds processes.

To verify which tools you have installed:

```
$ tfactl toolstatus
```

You can run each tool using `tfactl` either in command line or shell mode. To run a tool from the command line:

```
$ tfactl run tool
```

The following example shows how to use `tfactl` in shell mode. Running the command starts `tfactl`, connects to the database *MyDB*, and then runs `oratop`:

```
$ tfactl
tfactl > database MyDB
MyDB tfactl > oratop
```

Related Topics

- https://support.oracle.com/rs?type=doc&amp;id=2550798.1
- https://support.oracle.com/rs?type=doc&amp;id=1070954.1
- https://support.oracle.com/rs?type=doc&amp;id=2550798.1
- https://support.oracle.com/rs?type=doc&amp;id=301137.1
- https://support.oracle.com/rs?type=doc&amp;id=459694.1
- https://support.oracle.com/rs?type=doc&amp;id=1500864.1
- https://support.oracle.com/rs?type=doc&amp;id=215187.1

8.5 Searching Oracle Trace File Analyzer Metadata

You can search all metadata stored in the Oracle Trace File Analyzer index using `tfactl search -showdatatypes|-json [json_details]`.

You can search for all events for a particular Oracle Database between certain dates.

For example, on Linux systems:

```
tfactl search -json
'{
  "data_type":"event",
  "content":"oracle",
  "database":"rac11g",
  "from":"01/20/2017 00:00:00",
  "to":"12/20/2018 00:00:00"
 }'
```

For example, on Linux and Windows systems:

```
tfactl search -json
"{
   \"data_type\":\"event\",
   \"content\":\"oracle\",
   \"database\":\"rac11g\",
   \"from\":\"01/20/2017 00:00:00\",
   \"to\":\"12/20/2018 00:00:00\"
}"
```

To list all index events on Linux, AIX, and Solaris systems: `tfactl search -json '{"data_type":"event"}'`

To list all index events on Windows systems: `tfactl search -json "{\"data_type\":\"event\"}"`

To list all available datatypes: `tfactl search -showdatatypes`

8.6 Oracle Trace File Analyzer Service Request Data Collections (SRDCs)

Oracle Trace File Analyzer Service Request Data Collections (SRDCs) enable you to quickly collect the right diagnostic data.

To perform Service Request Data Collections:

```
$ tfactl diagcollect -srdc srdc_name
```

Running the command trims and collects all important log files updated in the past *n* hours across the whole cluster. The default number of hours for log collection varies from SRDC to SRDC. You can change the `diagcollect` timeframe with the `-last n h|d` option.

Oracle Support often asks you to run a Service Request Data Collection (SRDC). The SRDC depends on the type of problem that you experienced. An SRDC is a series of many data gathering instructions aimed at diagnosing your problem. Collecting the SRDC manually can be difficult with many different steps required.

Oracle Trace File Analyzer can run SRDC collections with a single command:

```
$ tfactl diagcollect
[-srdc srdc_profile]
[-sr sr_number]
[-tag tagname]
[-z filename]
[-last nh|d | -from time -to time | -for date]
[-database database]
```

Option	Description
`[-srdc srdc_profile]`	Specify the SRDC profile.
`-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.

<table>
<thead>
<tr><th>Option</th><th>Description</th></tr>
</thead>
<tbody>
<tr>
<td><code>[-last <i>nh|d</i> | -from <i>time</i> -to <i>time</i> | -for <i>date</i>]</code></td>
<td>
<ul>
<li>Specify the <code>-last</code> parameter to collect files that have relevant data for the past specific number of hours (<i>h</i>) or days (<i>d</i>). By default, using the command with this parameter also trims files that are large and shows files only from the specified interval.
You can also use <code>-since</code>, which has the same functionality as <code>-last</code>. This option is included for backward compatibility.</li>
<li>Specify the <code>-from</code> and <code>-to</code> parameters (you must use these two parameters together) to collect files that have relevant data during a specific time interval, and trim data before this time where files are large.
Supported time formats:
<code>"Mon/dd/yyyy hh:mm:ss"</code>
<code>"yyyy-mm-dd hh:mm:ss"</code>
<code>"yyyy-mm-ddThh:mm:ss"</code>
<code>"yyyy-mm-dd"</code></li>
<li>Specify the <code>-for</code> parameter to collect files that have relevant data for the date specified. The files <code>tfactl</code> collects will have timestamps in between which the time you specify after <code>-for</code> is included. No data trimming is done for this option.
Supported time formats:
<code>"Mon/dd/yyyy"</code>
<code>"yyyy-mm-dd"</code></li>
</ul>
<b>Note:</b>
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.
</td>
</tr>
<tr>
<td><code>-database <i>database</i></code></td>
<td>Specify the name of the database.</td>
</tr>
</tbody>
</table>

> **Note:**
>
> To upload collections to the SR as part of diag collection:
>
> If you have already set MOS configuration using the `tfactl setupmos` command, then you can use the `-sr` option along with the diag collection command. Note that `tfactl setupmos` is supported only in versions earlier than 20.2.
>
> If you have not set MOS configuration using the `tfactl setupmos` command, then set up MOS configuration using the new generic command, `ahfctl setupload -name mos -type https` and follow the instructions.
>
> For example: `tfactl diagcollect -srdc` *`srdc_type`* `-sr` *`sr_number`*

To run SRDCs, use one of the Oracle privileged user accounts:

- `ORACLE_HOME` owner
- `GRID_HOME` owner

Table 8-3 One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`ahf`	Oracle ORAchk and Oracle EXAchk problems (to be run after running with `-debug`)	Local only	No
`crs`	Collect `crs` traces	Cluster-wide	No
`crsasm`	ASM CRS-related problems	Cluster-wide	No
`crsasmcell`	ASM CRS CELL-related problems	Cluster-wide	No
`dbacl`	Problems with Access Control Lists (ACLs)	Local only	No
`dbaqgen`	Problems in an Oracle Advanced Queuing Environment	Local only	No
`dbaqmon`	Queue Monitor (QMON) problems	Local only	No
`dbaqnotify`	Notification problems in an Oracle Advanced Queuing Environment	Local only	No
`dbaqperf`	Performance problems in an Oracle Advanced Queuing Environment	Local only	No
`dbaqpurge`	Non-purged Messages in an Oracle Advanced Queuing Environment problems	Local only	No
`dbasm`	Oracle Database storage problems	Local only	No
`dbaudit`	Standard information for Oracle Database auditing	Local only	No
`dbaum`	AUM: Checklist of Evidence to Supply (Doc ID 1682741.1)	Local only	No
`dbaumwaitevents`	Wait Events related to Undo: Checklist of Evidence to Supply (Doc ID 1682723.1)	Local only	No
`dbawrspace`	Oracle Database Automatic Workload Repository (AWR) space problems	Local only	No
`dbbeqconnection`	Bequeath Connection Issues: Checklist of Evidence to Supply (Doc ID 1928047.1)	Local only	No
`dbcorrupt`	Generic Oracle Database corruption	Local only	No
`dbdataguard`	Data Guard problems including Broker	Local only	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
dbawrspace	Excessive SYSAUX space is used by the Automatic Workload Repository (AWR)	Local only	No
dbdatapatch	Datapatch problems	Local only	No
dbddlerrors	DDL Errors: Checklist of Evidence to Supply (Doc ID 2383662.1)	Local only	No
dbemon	Event Monitor (EMON) problems	Local only	No
dbenqdeq	Collect standard information for Advanced Queueing problems using TFA Collector (recommended) or manual steps	Local only	No
dbexp dbexpdp dbexpdpapi dbexpdpperf dbexpdptts	Original Oracle Database Export (exp)	Local only	No
dbfs	Oracle Automatic Storage Management (Oracle ASM) / Database File System (DBFS) / Direct NFS / Oracle Advanced Cluster File System (Oracle ACFS) problems	Local only	No
dbfra	Fast Recovery Area, also known as Flash Recovery Area problems	Local only	No
dbggclassicmode dbggintegratedmode	Oracle GoldenGate	Local only	No
dbhang	Oracle Database hang problems	Local only	No
dbhangperflite	Oracle Database performance and hang problems	Local only	No
dbimp dbimpdp dbimpdpperf	Original Oracle Database Import (imp)	Local only	No
dbimpdpperf	Data Pump Import performance problems	Local only	No
dbinstall dbupgrade dbpreupgrade	Oracle Database install / upgrade problems	Local only	No
dbparameters	Oracle Database single instance shutdown problems	Local only	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`dbparameterfiles`	Parameter Files: Checklist of Evidence to Supply (Doc ID 1914153.1)	Local only	No
`dbpartition`	Create or maintain partitioned table, subpartitioned table, and index problems	Local only	No
`dbpartitionperf`	Slow `Create`, `Alter`, or `Drop` commands against partitioned table or index	Local only	No
`dbpatchinstall` `dbpatchconflict`	Oracle Database patching problems	Local only	No
`dbperf`	Oracle Database performance problems	Cluster-wide	No
`dbperf_and_hang`	Oracle Database performance and hang problems on FASaaS environments	Local only	No
`dbplugincompliance`	Enterprise Manager compliance related issues	Local only	No
`dbpreupgrade`	Oracle Database preupgrade problems	Local only	No
`dbprocmgmt`	Generic Process Management and Related Issues: Checklist of Evidence to Supply (Doc ID 2500734.1)	Local only	No
`dbrac`	Oracle RAC-related data collection for Oracle Clusterware and Oracle ASM problems	Local only	No
`dbracinst`	Oracle RAC-related data collection for Oracle Database problems	Local only	No
`dbracperf`	Oracle RAC-related performance problems	Cluster-wide	No
`dbresmgr`	Oracle Database problems related to Resource Manager	Local only	No
`dbrman` `dbrmanperf`	Recovery Manager (RMAN) problems	Local only	No
`dbscn`	System change number	Local only	No
`dbshutdown` `dbstartup`	Oracle Database startup or shutdown problems	Local only	No
`dbslowddl`	Slow DDL: Checklist of Evidence to Supply	Local only	No
`dbspacewait`	Space Related Wait Events and Performance Issues : Checklist of Evidence to Supply (Doc ID 2560286.1)	Local only	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`dbspatialexportimport`	Oracle Spatial export or import problems	Local only	No
`dbspatialinstall`	Oracle Spatial installation problems	Local only	No
`dbspatialperf`	Oracle Spatial performance problems	Local only	No
`dbspatialupgrade`	Oracle Spatial upgrade problems	Local only	No
`dbspatialusage`	Oracle Spatial usage problems	Local only	No
`dbsqlperf`	SQL performance problems	Local only	No
`dbstandalonedbca`	Database Configuration Assistant problems	Local only	No
`dbstoragestructuregeneric`	Storage structure related diagnosis	Local only	No
`dbtablespacegeneric`	Generic Tablespace and Segment Management: Checklist of Evidence to Supply (Doc ID 2560291.1)	Local only	No
`dbtde`	Transparent Data Encryption (TDE) problems	Local only	No
`dbtextindex`	Oracle Text problems	Local only	No
`dbtextissue`	Oracle Text installation problems - 12c.	Local only	No
`dbtextupgrade` `dbtextinstall`	Oracle Text version 12.1.0.1 and later upgrade problems	Local only	No
`dbunixresources`	Oracle Database resource problems	Local only	No
`dbvault`	Collect standard information for Database Vault	Local only	No
`dbwindowsresources`	Oracle Database on Microsoft Windows resources: Checklist of Evidence to Supply	Local only	No
`dbwinservice`	OracleService on Microsoft Windows: Checklist of Evidence to Supply (Doc ID 1918781.1)	Local only	No
`dbxdb`	XDB Installation or invalid object problems	Local only	No
`dbxdbgeneric`	XDB installation and invalid object problems	Local only	No
`dbxdbupgrade`	XDB installation and invalid object problems in 12c and above	Local only	No
`dnfs`	XDB Upgrade problems	Local only	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`emagentgeneric`	Collect trace/log information for Enterprise Manager Management Agent generic problems	OMS and Agent	No
`emagentpatching`	Enterprise Manager failures during agent patching	OMS and Agent	No
`emagentperf`	Enterprise Manager 13c Agent performance problems	Agent	No
`emagentssl`	Enterprise Manager Agent SSL configuration issues	OMS and Agent	No
`emagentstartup`	Enterprise Manager 13c Agent startup problems	Agent	No
`emagentunreach`	Enterprise Manager 13c Agent unreachable errors or status	Agent	No
`emagentupload`	Enterprise Manager 13c Agent upload errors	Agent	No
`emagtpatchdeploy`	Enterprise Manager 13c Agent patch deployment problems	OMS and Agent	No
`emagtupginst`	Collecting diagnostic data for Enterprise Manager 13c Agent installation, upgrade, or deployment prodblems	Agent	No
`emagtupgpatch`	Enterprise Manager 13c Agent upgrade, local installation, or patching problems.	OMS and Agent	No
`emauthldap`	Enterprise Manager authentication using LDAP provider issues	OMS	No
`emblackout`	Enterprise Manager Blackout Issues	OMS and Agent	No
`emcliadd`	Enterprise Manager target discovery or add problems	OMS and Agent	No
`emclusdisc`		Agent	
`emdbsys`		OMS and Agent	
`emgendisc`		OMS and Agent	
`emprocdisc`		OMS and Agent	
`emcomm`	Enterprise Manager communication information between OMS and Agent	OMS and Agent	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`emdbaasdeploy`	Database As A Service (DBaaS): Collect trace or log information for failures during DBaaS deployment.	OMS and remote DBaaS deployment server	No
`emdebugon` `emdebugoff`	Enterprise Manager debug log collection Run `emdebugon`, reproduce the problem then run `emdebugoff`, which disables debug again and collects debug logs	OMS or Agent	No
`emfleetpatching`	Enterprise Manager Fleet Maintenance Patching problems	OMS and Agent	No
`emjobs`	Enterprise Manager all job related issues	OMS	No
`emmetricalert`	Enterprise Manager general metrics page or threshold problems	Agent	No
`emnotif`	Enterprise Manager incident rules and notification issues	OMS	No
`emomsfailstart`	Enterprise Manager OMS startup failures	OMS	No
`emomscrash`	Enterprise Manager OMS Crash problems	OMS	No
`emomsheap`	Enterprise Manager Java heap usage or performance problems	OMS	No
`emomshungcpu`	Enterprise Manager OMS crash, restart or performance problems	OMS	No
`emomspatching`	Enterprise Manager failures during OMS patching	OMS	No
`emomsssl`	Enterprise Manager OMS SSL configuration issues	OMS	No
`emomsupginst`	Enterprise Manager OMS installation, upgrade, and patching	Local only	No
`empatchplancrt`	Enterprise Manager patch plan creation problems	OMS and Agent	No
`emprocdisc`	Oracle Database, Listener, or ASM target is not discovered or detected by the discovery process	Local only	No
`emtbsmetrics`	Enterprise Manager tablespace usage metric problems	Local only (on Enterprise Manager Agent target)	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection
`emwlsssl`	Enterprise Manager WebLogic Server (WLS) SSL configuration issues	Local only	No
`esexalogic`	Oracle Exalogic full Exalogs data collection information	Local only	No
`exadata`	Collect Oracle Exadata information	Local only	No
`exservice`	Oracle Exadata: Storage software service or offload server service problems	Local only	No
`exsmartscan`	Oracle Exadata: Smart Scan not working problems	Local only	No
`generic`	Fallthrough SRDC for Oracle Database error	Local only	No
`gg_abend`	Oracle GoldenGate covering both Classic and Microservices implementations.	Local only	No
`ggintegratedmodenodb`	Oracle GoldenGate extract/ replicate abends problems.	Local only	No
`gridinfra`	Oracle RAC-related data collection for Oracle Clusterware problems	Local only	No
`gridinfrainst`	Oracle RAC upgrade and patching problems	Local only	No
`instterm`	Collect traces for the following ORA errors: • `ORA-00469` • `ORA-00470` • `ORA-00480` • `ORA-00490` • `ORA-00491` • `ORA-00492` • `ORA-00493` • `ORA-00495` • `ORA-00496` • `ORA-00497` • `ORA-00498`	Local only	No
`internalerror`	Other internal Oracle Database errors	Local only	No
`listener_services`	Listener errors: `TNS-12516` / `TNS-12518` / `TNS-12519` / `TNS-12520`	Local only	No
`naming_services`	Naming service errors: `TNS-12154` / `TNS-12528`	Local only	No

Table 8-3 (Cont.) One Command Service Request Data Collections

Available SRDCs	Type of Problem	Collection Scope	Auto Collection

`ORA-00020`
`ORA-00060`
`ORA-00494`
`ORA-00600`
`ORA-00700`
`ORA-01031`
`ORA-01555`
`ORA-01578`
`ORA-01628`
`ORA-03137`
`ORA-04020`
`ORA-04021`
`ORA-04030`
`ORA-04023`
`ORA-04031`
`ORA-04063`
`ORA-07445`
`ORA-08102`
`ORA-08103`
`ORA-22924`
`ORA-27300`
`ORA-27301`
`ORA-27302`
`ORA-30036`
	ORA Errors	Local only	Only the following SRDCs:
			• `ORA-00600`
			• `ORA-04030`
			• `ORA-04031`
			• `ORA-04021`
			• `ORA-07445`
			• `ORA-01578`

`ORA-01000`	Open Cursors problems	Local only	No
`ORA-00018`	`ORA-00018` or sessions parameter problems	Local only	No
`ORA-12751`	`ORA-12751` collection errors	Local only	No
`ORA-25319`	Collect information for troubleshooting `ORA-25319` error in an Advanced Queuing Environment `ORA-25319` problems in an Oracle Advanced Queuing Environment	Local only	No
`ORA-00227`	Collect information for troubleshooting Control File block corruption reported by error `ORA-00227`	Local only	No
`privsroles`	Data Collection for privileges and roles	Local only	No
`xdb600`	Diagnostic data collection for XDB `ORA-00600` and `ORA-07445` internal rrror issues using TFA Collector	Local only	No
`zlgeneric`	Zero Data Loss Recovery Appliance (ZDLRA) problems	Local only	No

For more information about SRDCs, run `tfactl diagcollect -srdc -help`.

The types of information that the SRDCs collect varies for each type, for example, the following table lists and describes what the SRDCs collect for each type.

Table 8-4 SRDC collections

Command	What gets collected
`$ tfactl diagcollect -srdc ORA-04031`	• Incident Packaging Service (IPS) package • Patch listing • Automatic Workload Repository (AWR) report • Memory information
`$ tfactl diagcollect -srdc dbperf`	• Automatic Database Diagnostic Monitor (ADDM) report • Automatic Workload Repository (AWR) for good period and problem period • Automatic Workload Repository (AWR) Compare Period report • Active Session History (ASH) report for good and problem period • OSWatcher • Incident Packaging Service (IPS) package (if there are any errors during problem period) • Oracle ORAchk (performance-related checks)

Oracle Trace File Analyzer prompts you to enter the information required based on the SRDC type.

For example, when you run `ORA-4031` SRDC:

```
$ tfactl diagcollect -srdc ORA-04031
```

Oracle Trace File Analyzer:

1. Prompts to enter event date, time, and database name.
2. Scans the system to identify recent events in the system (up to 10).
3. Proceeds with diagnostic collection after you choose the relevant event.
4. Identifies all the required files.
5. Trims all the files where applicable.
6. Packages all data in a zip file ready to provide to support.

You can also run an SRDC collection in non-interactive silent mode. Provide all the required parameters up front as follows:

```
$ tfactl diagcollect -srdc srdc_type -database db -from "date time" -to "date
time"
```

Related Topics

- https://support.oracle.com/rs?type=doc&id=1918781.1
- https://support.oracle.com/rs?type=doc&id=2560291.1
- https://support.oracle.com/rs?type=doc&id=2560286.1
- https://support.oracle.com/rs?type=doc&id=2500734.1
- https://support.oracle.com/rs?type=doc&id=1914153.1

- https://support.oracle.com/rs?type=doc&id=2383662.1
- https://support.oracle.com/rs?type=doc&id=1682741.1
- https://support.oracle.com/rs?type=doc&id=1682723.1
- https://support.oracle.com/rs?type=doc&id=1928047.1

8.7 Diagnostic Upload

Diagnostic upload eliminates the need for different set of commands to upload Oracle ORAchk, Oracle EXAchk, and Oracle Trace File Analyzer diagnostic collections to AHF Service, database, and Oracle Support.

Diagnostic upload enables you to manage configurations of different types of uploads in a generic way. Through `ahfctl` command-line interface, you use generic upload commands to set, get, unset, and check configurations. Configurations are uniquely identified using configuration name so that you can pass the configuration name in command-line to perform upload and other operations.

AHF synchronizes the configuration automatically across the cluster nodes. If you find any sync issues, then run the `tfactl syncahfconfig -c` command to sync configuration across the cluster nodes.

Diagnostic upload supports multiple operating system users to run the diagnostic upload commands if you install AHF as `root`. If you install AHF as a non-root user, then you cannot benefit from the multiple operating system users support.

> **Note:**
>
> Currently not supported on Microsoft Windows.

Currently, AHF supports HTTP, SQLNET, and SFTP types or protocols, or end points. Following sections list the parameters or arguments supported by different end points while setting the configuration.

HTTP

Set Parameters: `url`, `user`, `password`, `proxy`, `noauth`, `https_token`, `header`, `secure`, and `storetype`

Upload Parameters: `id`, `file`, and `https_token`

SQLNET

Set Parameters: `user`, `password`, `connectstring`, and `uploadtable`

Upload Parameters: `file`

SFTP

Set Parameters: `server`, `user`, and `password`

Upload Parameters: (optional) `id` and `file`

Parameters or arguments Supported by Different Endpoints

Table 8-5 Parameters or Arguments Supported by Different Endpoints

Parameter	Description
`url`	The target URL to upload files in case of HTTPS type. For example, *https://samplehost.com*.
`server`	The name of the server to which you want to upload files. For example, *sftpserver.domain.com*.
`user`	The user who has the privileges to access the endpoint. For example, *upload.user@example.com*.
`password`	Password of the user.
`proxy`	The URL of the proxy server. For example, *www.example.com:80*.
`id`	The location or target where you want to upload your files to.
`file`	The name of the file to upload.
`noauth`	Specify `true` and `false`. Default value is `false`. If `noauth` is set to `true`, then HTTPS upload will skip authentication. For example, upload files to PAR, Pre Authenticated URL where no user/password authentication is required.
`https_token`	Any static header values while configuring. For example, set auth tokens while configuring the HTTPS end point. For example, `ahfctl setupload -name config -type https -https_token 'abc:13'`. You can also pass dynamic headers at upload time by passing the `-https_token headers` command option to `tfactl upload` command. For example: `-H 'X-TFA-REQUESTID: 1'`.
`header`	Stores the `executionId` in the `ahf.properties` file. For example, to set the header:`ahfctl setupload -name a1 -type https -header X-TFA-HEADERS:executionId=aeldb1db01_2020.06.16_19.20.55.15336025`
`secure`	Specify `true` or `false`. Default value is `true`. Specifying the secure value checks for certificates. If `secure` is set to `false`, then the `upload` command will run an unsecure upload.
`connectstring`	The database connect string to log in to the database where you want to upload files. For example, `(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME = orcl)))`.

Table 8-5 (Cont.) Parameters or Arguments Supported by Different Endpoints

Parameter	Description
`uploadtable`	Specify the name of the table where you want to upload files as `BLOB` type. For example, for uploading Oracle ORAchk collections to the Collection Manager it is set to `RCA13_DOCS`.

Example 8-1 Diagnostic Upload Examples To upload files to My Oracle Support

To setup MOS configuration:

```
ahfctl setupload -name mos -type https

Enter mos.https.user: user_id
Enter mos.https.password: ########
Enter mos.https.url: https://transport.oracle.com/upload/issue

Upload configuration set for: mos
type: https
user: user_id
password: ########
url: https://transport.oracle.com/upload/issue
```

To set proxy:

```
ahfctl setupload -name mos -type https -proxy www-proxy.example.com:80
```

Single-line command:

```
ahfctl setupload -name mos -type https -user user_id -url https://
transport.oracle.com/upload/issue -proxy www-proxy.example.com:80
```

> **Note:**
>
> Instead of `mos`, you can specify any configuration name of your choice.

To upload collections or files to MOS: There are multiple ways you can upload files to MOS after configuring MOS.

- Upload files as part of Oracle Trace File Analyzer diagnostic collection:

  ```
  tfactl diagcollect -last 1h -upload mos -id 3-23104325631
  ```

- Upload files standalone:

  ```
  tfactl upload -name mos -id 3-23104325631 -file /tmp/generated.zip
  ```

- Backward compatibility or upload using `-sr` flag with `diagcollcet` command:

  ```
  tfactl diagcollect -last 1h -sr 3-23104325631
  ```

> **Note:**
>
> In this case, upload configuration name should be `mos` as internally Oracle Trace File Analyzer looks for this name. It works even if MOS configuration is set using the `tfactl setupmos` command in versions earlier than 20.2.

Example 8-2   Uploading a File Using SFTP

```
ahfctl upload -name sftp1 -file test_sftp_upload.log
Upload for: sftp1
Uploading file using pexpect
sftp> put test_sftp_upload.log
put test_sftp_upload.log
Uploading test_sftp_upload.log to /root/test_sftp_upload.log
test_sftp_upload.log                          100%    17     0.0KB/s
00:00
sftp> quit
type: sftp
file: test_sftp_upload.log
Upload completed successfully
```

Example 8-3   Diagnostic Upload Examples

To set configuration parameters for the specified configuration name and SQLNET configuration type:

```
ahfctl setupload -name mysqlnetconfig -type sqlnet
```

```
[root@myserver1]# ahfctl setupload -name mysqlnetconfig -type sqlnet
Enter mysqlnetconfig.sqlnet.user: testuser
Enter mysqlnetconfig.sqlnet.password: ########
Enter mysqlnetconfig.sqlnet.connectstring: (DESCRIPTION = (ADDRESS =
(PROTOCOL = TCP)(HOST = testhost)(PORT = 1521))(CONNECT_DATA =(SERVER =
DEDICATED)(SERVICE_NAME = testservice)))
Enter mysqlnetconfig.sqlnet.uploadtable: RCA13_DOCS

Upload configuration set for: mysqlnetconfig
type: sqlnet
user: testuser
password: ########

connectstring: (DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = testhost)
(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME =
testservice)))
uploadtable: RCA13_DOCS
```

To set individual parameters for the specified configuration name and SQLNET configuration type:

```
ahfctl setupload -name mysqlnetconfig2 -type sqlnet -user user_name@example.com
```

This omits the `-password` option and therefore reports:

```
Database upload parameter(s) successfully stored.
AHF will not upload collections into the database until the following
parameters are also set:
['password', 'connectstring', 'uploadtable']
```

When you specify the `-user` command option, `ahfctl` does NOT prompt for the other required parameters so you must explicitly specify them at the command line as follows:

```
ahfctl setupload -type sqlnet -name orachkcm -user testuser -password  -
connectstring sqlnet connect string -uploadtable RCA13_DOCS
```

The `-password` command option DOES NOT take any arguments. When specified, `ahfctl` prompts you to provide the password for the user you specified using the `-user` command option.

To get the list of all configured names in the `AHF.properties` file:

```
ahfctl getupload
```

```
# ahfctl getupload
Upload configurations available:
1. mysftpconfig
2. myhttpsconfig
3. mysqlnetconfig
```

To get all configuration parameters for the specified configuration name:

```
ahfctl getupload -name mysftpconfig
```

```
# ahfctl getupload -name mysftpconfig
Upload configuration get for: mysftpconfig
type: sftp
user: testuser1@example.com
password: #########
server: sftphost.example.com
```

To get individual parameter for the specified configuration name:

```
ahfctl getupload -name mysftpconfig -user
```

```
[root@myserver1]# ahfctl getupload -name mysftpconfig -user
Upload configuration get for: mysftpconfig
type: sftp
user: testuser1@example.com
```

To check or validate configuration of the specified configuration name:

```
ahfctl checkupload -name mysftpconfig
```

```
# ahfctl checkupload -name mysftpconfig -type sftp
Upload configuration check for: mysftpconfig
Parameters are configured correctly to upload files to sftp end point
mysftpconfig
```

To upload to target using the configuration name specified:

```
tfactl upload -name mysftpconfig -id 30676598 -file /tmp/temp.txt
```

```
# tfactl upload -name mysftpconfig -id 30676598 -file /tmp/filename.txt
Upload for: mysftpconfig
type: sftp
file: /tmp/filename.txt
id: 30676598
Upload completed successfully.
```

To unset individual parameter of the specified configuration name:

```
ahfctl unsetupload -name mysftpconfig -user
```

```
# ahfctl unsetupload -name mysftpconfig -user
Upload configuration successfully unset for: mysftpconfig
```

To unset all parameters of the specified configuration name:

```
ahfctl unsetupload -name mysftpconfig -all
```

```
# ahfctl unsetupload -name mysftpconfig -all
Upload configuration successfully unset for: mysftpconfig
```

To auto upload generated zip file to the database using Oracle ORAchk:

```
exachk -showpass -localonly -check BF7AE780E1252F69E0431EC0E50AE447
```

```
# exachk -showpass -localonly -check BF7AE780E1252F69E0431EC0E50AE447
Orachk.zip successfully uploaded to RCA13_DOCS table
```

To auto upload generated zip file to MOS using `tfactl diagcollect`:

```
$ tfactl diagcollect -since 1h -upload mos -id 3-123456789
```

To upload generated zip to the database with the configurations set by AHF with the specified database config name:

```
exachk -showpass -localonly -check BF7AE780E1252F69E0431EC0E50AE447 -
db_config_name user_dbconf
```

```
# exachk -showpass -localonly -check BF7AE780E1252F69E0431EC0E50AE447 -
db_config_name user_dbconf
Orachk.zip successfully uploaded to RCA13_DOCS table
```

8.8 Changing Oracle Grid Infrastructure Trace Levels

Enable trace levels to collect enough diagnostics to diagnose the cause of the problem.

Oracle Support asks you to enable certain trace levels when reproducing a problem. You can enable and then disable the trace levels. Use the `dbglevel` option to set the trace level. You can find the required trace level settings grouped by problem trace profiles.

To set trace levels:

1. To set a trace profile:

   ```
   tfactl dbglevel –set profile
   ```

2. To list all available profiles:

   ```
   tfactl dbglevel –help
   ```

- tfactl dbglevel
  Use the `tfactl dbglevel` command to set Oracle Grid Infrastructure trace levels.

8.8.1 tfactl dbglevel

Use the `tfactl dbglevel` command to set Oracle Grid Infrastructure trace levels.

Syntax

```
tfactl [run] dbglevel
[ {-set|-unset} profile_name
-dependency [dep1,dep2,...|all]
-dependency_type [type1,type2,type3,...|all]
| {-view|-drop} profile_name
| -lsprofiles
| -lsmodules
| -lscomponents [module_name]
| -lsres
| -create profile_name [ -desc description
| [-includeunset] [-includetrace]
| -debugstate | -timeout time ]
| -modify profile_name [-includeunset] [-includetrace]
| -getstate [ -module module_name ]
| -active [profile_name]
| -describe [profile_name] ] ]
```

Parameters

Table 8-6 tfactl dbglevel Command Parameters

Parameter	Description
`profile_name`	Specify the name of the profile.
`active`	Displays the list of active profiles.
`set`	Sets the trace or log levels for the profile specified.
`unset`	Unsets the trace or log levels for the profile specified.
`view`	Displays the trace or log entries for the profile specified.
`create`	Creates a profile.
`drop`	Drops the profile specified.
`modify`	Modifies the profile specified.
`describe`	Describes the profiles specified.
`lsprofiles`	Lists all the available profiles.
`lsmodules`	Lists all the discovered Oracle Clusterware modules.
`lscomponents`	Lists all the components associated with the Oracle Clusterware module.
`lsres`	Lists all the discovered Oracle Clusterware resources.
`getstate`	Displays the current trace or log levels for the Oracle Clusterware components or resources.
`module`	Specify the Oracle Clusterware module.
`dependency`	Specify the dependencies to consider, start, or stop dependencies, or both.
`dependency_type`	Specify the type of dependencies to be consider.
`debugstate`	Generates a System State Dump for all the available levels.
`includeunset`	Adds or modifies an unset value for the Oracle Clusterware components or resources.
`includetrace`	Adds or modifies a trace value for the Oracle Clusterware components.

> **WARNING:**
>
> Set the profiles only at the direction of Oracle Support.

8.9 Performing Custom Collections

Use the custom collection options to change the diagnostic collections from the default.

- Adjusting the Diagnostic Data Collection Period
  Oracle Trace File Analyzer trims and collects any important logs updated in the past one hour.

- Collecting for Specific Events
  Perform default diagnostic collection or choose an event from the list of recent incidents to collect diagnostic data for that event alone.
- Excluding Large Files from Diagnostic Collection
  Prevent excessively large files from delaying or stalling collections.
- Collecting from Specific Nodes
- Collecting from Specific Components
- Collecting from Specific Directories
- Changing the Collection Name
- Preventing Copying Zip Files and Trimming Files
- Performing Silent Collection
- Collecting Core Files
- Collecting Incident Packaging Service (IPS) Packages
  Incident Packaging Service packages details of problems stored by Oracle Database in ADR for later diagnosis.

8.9.1 Adjusting the Diagnostic Data Collection Period

Oracle Trace File Analyzer trims and collects any important logs updated in the past one hour.

If you know that you only want logs for a smaller window, then you can cut this collection period. Cutting the collection period helps you make collections as small and quick as possible.

There are four different ways you can specify the period for collection:

Table 8-7 Ways to Specify the Collection Period

Command	Description
`tfactl diagcollect -last` *n* `h|d`	Collects since the previous *n* hours or days. • Number of days must be less than or equal to 7 • Number of hours must be less than or equal to 168

`tfactl diagcollect -from “`*yyyy-mm-dd*`”`	Collects from the date and optionally time specified.
	Valid date and time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`

`tfactl diagcollect -from "`*yyyy-mm-dd*`" -to "`*yyyy-mm-dd*`"`	Collects between the date and optionally time specified.
	Valid date and time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`

`tfactl diagcollect -for “`*yyyy-mm-dd*`”`	Collects for the specified date.
	Valid date formats:
	`"Mon/dd/yyyy"`
	`"yyyy-mm-dd"`

8.9.2 Collecting for Specific Events

Perform default diagnostic collection or choose an event from the list of recent incidents to collect diagnostic data for that event alone.

Choose to run:

- A diagnostic collection for a specific recent event
- A default time range diagnostic collection

To collect for specific events:

1. To run a default diagnostic collection:

   ```
   tfactl diagcollect
   ```

   For example:

   ```
   $ tfactl diagcollect
   Choose the event you want to perform a diagnostic collection for:
   1. Mar/12/2019 16:08:20 [ db.orcl.orcl ]  ORA-04030: out of process memory
   when trying to allocate
   2. Mar/12/2019 16:08:18 [ db.orcl.orcl ]  ORA-04031: unable to allocate 8
   bytes of shared memory
   3. Mar/12/2019 16:08:16 [ db.orcl.orcl ]  ORA-00494: enqueue held for too
   long more than seconds by osid
   4. Mar/12/2019 16:08:14 [ db.orcl.orcl ]  ORA-29709: Communication failure
   with Cluster Synchronization
   5. Mar/12/2019 16:08:04 [ db.orcl.orcl ]  ORA-29702: error occurred in
   Cluster Group Service operation
   6. Mar/12/2019 16:07:59 [ db.orcl.orcl ]  ORA-32701: Possible hangs up to
   hang ID= detected
   7. Mar/12/2019 16:07:51 [ db.orcl.orcl ]  ORA-07445: exception
   encountered: core dump [] [] [] [] [] []
   8. Mar/12/2019 16:07:49 [ db.orcl.orcl ]  ORA-00700: soft internal error,
   arguments: [700], [], [],[]
   9. Mar/11/2019 22:02:19 [ db.oradb.oradb ]  DIA0 Critical Database Process
   Blocked: Hang ID 1 blocks 5 sessions
   10. Default diagnostic collection, for no specific event

   Please choose the event : 1-10 [] 10

   By default TFA will collect diagnostics for the last 12 hours. This can
   result in large collections
   For more targeted collections enter the time of the incident, otherwise
   hit <RETURN> to collect for the last 12 hours
   [YYYY-MM-DD HH24:MI:SS,<RETURN>=Collect for last 12 hours] :

   Collecting data for the last 12 hours for all components...
   Collecting data for all nodes

   Collection Id : 20190312163846node1

   Detailed Logging at : /scratch/app/product/19c/tfa/repository/
   ```

```
collection_Tue_Mar_12_16_38_47_PDT_2019_node_all/
diagcollect_20190312163846_node1.log
2019/03/12 16:38:50 PDT : NOTE : Any file or directory name containing the
string .com will be renamed to replace .com with dotcom
2019/03/12 16:38:50 PDT : Collection Name :
tfa_Tue_Mar_12_16_38_47_PDT_2019.zip
2019/03/12 16:38:50 PDT : Collecting diagnostics from hosts : [node1]
2019/03/12 16:38:50 PDT : Scanning of files for Collection in progress...
2019/03/12 16:38:50 PDT : Collecting additional diagnostic information...
2019/03/12 16:38:55 PDT : Getting list of files satisfying time range
[03/12/2019 04:38:50 PDT, 03/12/2019 16:38:55 PDT]
2019/03/12 16:39:02 PDT : Collecting ADR incident files...
2019/03/12 16:39:06 PDT : Completed collection of additional diagnostic
information...
2019/03/12 16:39:07 PDT : Completed Local Collection
.-------------------------------------.
|         Collection Summary          |
+----------+-----------+------+------+
| Host     | Status    | Size | Time |
+----------+-----------+------+------+
| node1 | Completed | 21MB |  17s |  |
'----------+-----------+------+------'

Logs are being collected to: /scratch/app/product/19c/tfa/repository/
collection_Tue_Mar_12_16_38_47_PDT_2019_node_all
/scratch/app/product/19c/tfa/repository/
collection_Tue_Mar_12_16_38_47_PDT_2019_node_all/
node1.tfa_Tue_Mar_12_16_38_47_PDT_2019.zip


$ tfactl diagcollect
Choose the event you want to perform a diagnostic collection for:
1. Mar/12/2019 16:08:20 [ db.orcl.orcl ]  ORA-04030: out of process memory
when trying to allocate
2. Mar/12/2019 16:08:18 [ db.orcl.orcl ]  ORA-04031: unable to allocate 8
bytes of shared memory
3. Mar/12/2019 16:08:16 [ db.orcl.orcl ]  ORA-00494: enqueue held for too
long more than seconds by osid
4. Mar/12/2019 16:08:14 [ db.orcl.orcl ]  ORA-29709: Communication failure
with Cluster Synchronization
5. Mar/12/2019 16:08:04 [ db.orcl.orcl ]  ORA-29702: error occurred in
Cluster Group Service operation
6. Mar/12/2019 16:07:59 [ db.orcl.orcl ]  ORA-32701: Possible hangs up to
hang ID= detected
7. Mar/12/2019 16:07:51 [ db.orcl.orcl ]  ORA-07445: exception
encountered: core dump [] [] [] [] [] []
8. Mar/12/2019 16:07:49 [ db.orcl.orcl ]  ORA-00700: soft internal error,
arguments: [700], [], [],[]
9. Mar/11/2019 22:02:19 [ db.oradb.oradb ]  DIA0 Critical Database Process
Blocked: Hang ID 1 blocks 5 sessions
10. Default diagnostic collection, for no specific event

Please choose the event : 1-10 [] 1
User root does not have permissions to run SRDC 'ora4030' for database
'orcl'.
```

2. To run a diagnostic collection for a specific event that does not have an SRDC:

```
tfactl diagcollect
```

For example:

```
$ tfactl diagcollect
Choose the event you want to perform a diagnostic collection for:
1. Mar/12/2019 16:08:20 [ db.orcl.orcl ]  ORA-04030: out of process memory
when trying to allocate
2. Mar/12/2019 16:08:18 [ db.orcl.orcl ]  ORA-04031: unable to allocate 8
bytes of shared memory
3. Mar/12/2019 16:08:16 [ db.orcl.orcl ]  ORA-00494: enqueue held for too
long more than seconds by osid
4. Mar/12/2019 16:08:14 [ db.orcl.orcl ]  ORA-29709: Communication failure
with Cluster Synchronization
5. Mar/12/2019 16:08:04 [ db.orcl.orcl ]  ORA-29702: error occurred in
Cluster Group Service operation
6. Mar/12/2019 16:07:59 [ db.orcl.orcl ]  ORA-32701: Possible hangs up to
hang ID= detected
7. Mar/12/2019 16:07:51 [ db.orcl.orcl ]  ORA-07445: exception
encountered: core dump [] [] [] [] [] []
8. Mar/12/2019 16:07:49 [ db.orcl.orcl ]  ORA-00700: soft internal error,
arguments: [700], [], [],[]
9. Mar/11/2019 22:02:19 [ db.oradb.oradb ]  DIA0 Critical Database Process
Blocked: Hang ID 1 blocks 5 sessions
10. Default diagnostic collection, for no specific event

Please choose the event : 1-10 [] 9
Collecting data for all nodes
Scanning files from mar/11/2019 18:02:19 to mar/11/2019 23:02:19

Collection Id : 20190312162708node1

Detailed Logging at : /scratch/app/product/19c/tfa/repository/
collection_Tue_Mar_12_16_27_09_PDT_2019_node_all/
diagcollect_20190312162708_node1.log
2019/03/12 16:27:12 PDT : NOTE : Any file or directory name containing the
string .com will be renamed to replace .com with dotcom
2019/03/12 16:27:12 PDT : Collection Name :
tfa_Tue_Mar_12_16_27_09_PDT_2019.zip
2019/03/12 16:27:12 PDT : Collecting diagnostics from hosts : [node1]
2019/03/12 16:27:12 PDT : Scanning of files for Collection in progress...
2019/03/12 16:27:12 PDT : Collecting additional diagnostic information...
2019/03/12 16:27:17 PDT : Getting list of files satisfying time range
[03/11/2019 18:02:19 PDT, 03/11/2019 23:02:19 PDT]
2019/03/12 16:27:23 PDT : Collecting ADR incident files...
2019/03/12 16:27:28 PDT : Completed collection of additional diagnostic
information...
2019/03/12 16:27:33 PDT : Completed Local Collection
.-------------------------------------.
|         Collection Summary          |
+-----------+-----------+------+------+
| Host      | Status    | Size | Time |
+-----------+-----------+------+------+
| node1 | Completed | 10MB |  21s |
'----------+-----------+------+------'

Logs are being collected to: /scratch/app/product/19c/tfa/repository/
collection_Tue_Mar_12_16_27_09_PDT_2019_node_all
/scratch/app/product/19c/tfa/repository/
collection_Tue_Mar_12_16_27_09_PDT_2019_node_all/
node1.tfa_Tue_Mar_12_16_27_09_PDT_2019.zip
```

3. To run a diagnostic collection for a specific event that has an SRDC:

> **Note:**
>
> When choosing an SRDC the user running the collection needs to be in the dba group of the database chosen in the event.

```
tfactl diagcollect
```

For example:

```
$ tfactl diagcollect
Choose the event you want to perform a diagnostic collection for:
1. Mar/12/2019 16:08:20 [ db.orcl.orcl ]  ORA-04030: out of process memory
when trying to allocate
2. Mar/12/2019 16:08:18 [ db.orcl.orcl ]  ORA-04031: unable to allocate 8
bytes of shared memory
3. Mar/12/2019 16:08:16 [ db.orcl.orcl ]  ORA-00494: enqueue held for too
long more than seconds by osid
4. Mar/12/2019 16:08:14 [ db.orcl.orcl ]  ORA-29709: Communication failure
with Cluster Synchronization
5. Mar/12/2019 16:08:04 [ db.orcl.orcl ]  ORA-29702: error occurred in
Cluster Group Service operation
6. Mar/12/2019 16:07:59 [ db.orcl.orcl ]  ORA-32701: Possible hangs up to
hang ID= detected
7. Mar/12/2019 16:07:51 [ db.orcl.orcl ]  ORA-07445: exception
encountered: core dump [] [] [] [] [] []
8. Mar/12/2019 16:07:49 [ db.orcl.orcl ]  ORA-00700: soft internal error,
arguments: [700], [], [],[]
9. Mar/11/2019 22:02:19 [ db.oradb.oradb ]  DIA0 Critical Database Process
Blocked: Hang ID 1 blocks 5 sessions
10. Default diagnostic collection, for no specific event

Please choose the event : 1-10 [] 1
Scripts to be run by this srdc: srdc_db_sid_memorysizes_10glower.sql
srdc_db_sid_memorysizes_11gplus.sql ipspack
Components included in this srdc: OS DATABASE CHMOS
Collecting data for local node(s)
Scanning files from Mar/12/2019 14:08:20 to Mar/12/2019 18:08:20
WARNING: End time entered is after the current system time.

Collection Id : 20190312163524node1

Detailed Logging at : /scratch/app/product/19c/tfa/repository/
```

```
srdc_ora4030_collection_Tue_Mar_12_16_35_25_PDT_2019_node_local/
diagcollect_20190312163524_node1.log
2019/03/12 16:35:30 PDT : NOTE : Any file or directory name containing the
string .com will be renamed to replace .com with dotcom
2019/03/12 16:35:30 PDT : Collection Name :
tfa_srdc_ora4030_Tue_Mar_12_16_35_25_PDT_2019.zip
2019/03/12 16:35:30 PDT : Scanning of files for Collection in progress...
2019/03/12 16:35:30 PDT : Collecting additional diagnostic information...
2019/03/12 16:35:35 PDT : Getting list of files satisfying time range
[03/12/2019 14:08:20 PDT, 03/12/2019 16:35:30 PDT]
2019/03/12 16:35:49 PDT : Collecting ADR incident files...
2019/03/12 16:35:52 PDT : Completed collection of additional diagnostic
information...
2019/03/12 16:35:54 PDT : Completed Local Collection
.--------------------------------------.
|          Collection Summary          |
+----------+-----------+-------+------+
| Host     | Status    | Size  | Time |
+----------+-----------+-------+------+
| node1 | Completed | 2.9MB |  24s |
'----------+-----------+-------+------'

Logs are being collected to: /scratch/app/product/19c/tfa/repository/
srdc_ora4030_collection_Tue_Mar_12_16_35_25_PDT_2019_node_local
/scratch/app/product/19c/tfa/repository/
srdc_ora4030_collection_Tue_Mar_12_16_35_25_PDT_2019_node_local/
node1.tfa_srdc_ora4030_Tue_Mar_12_16_35_25_PDT_2019.zip
```

8.9.3 Excluding Large Files from Diagnostic Collection

Prevent excessively large files from delaying or stalling collections.

Run the `tfactl set` *`maxfilecollectionsize`* for the diagnostic collection command to consider only the last 200 KB for the files that are larger than the size specified.

1. To set the maximum file size:

   ```
   tfactl set maxfilecollectionsize=size_in_MB
   ```

2. To collect diagnostic data:

   ```
   tfactl diagcollect
   ```

8.9.4 Collecting from Specific Nodes

To collect from specific nodes:

- To collect from specific nodes:

  ```
  tfactl diagcollect -node list of nodes
  ```

For example:

```
$ tfactl diagcollect -last 1d -node myserver65
```

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.5 Collecting from Specific Components

To collect from specific components:

- To collect from specific components:

  ```
  tfactl diagcollect component
  ```

  For example:

  To trim and collect all files from the databases *hrdb* and *fdb* in the last 1 day:

  ```
  $ tfactl -diagcollect -database hrdb,fdb -last 1d
  ```

  To trim and collect all Oracle Clusterware files, operating system logs, and CHMOS/OSW data from *node1* and *node2* updated in the last 6 hours:

  ```
  $ tfactl diagcollect -crs -os -node node1,node2 -last 6h
  ```

  To trim and collect all Oracle ASM logs from *node1* updated between from and to time:

  ```
  $ tfactl diagcollect -asm -node node1 -from "2016-08-15" -to "2016-08-17"
  ```

  Following are the available component options.

Table 8-8 Component Options

Component Option	Description
`-cha`	Collects Oracle Cluster Health Advisor logs.
`-ips`	Collects Incident Packaging Service logs.
`-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 Database Workload Management (DBWLM) logs.
`-tns`	Collects TNS logs.
`-rhp`	Collects Rapid Home Provisioning (RHP) logs.
`-procinfo`	Collects `Gathers stack` and `fd` from `/proc` for all processes.

Table 8-8 (Cont.) Component Options

Component Option	Description
`-afd`	Collects AFD logs.
`-crs`	Collects Oracle Clusterware logs.
`-wls`	Collects Oracle WebLogic Server (WLS) logs.
`-emagent`	Collects Oracle Enterprise Manager Agent (EMAGENT) logs.
`-oms`	Collects Oracle Management Service (OMS) logs.
`-ocm`	Collects Oracle Configuration Manager (OCM) logs.
`-emplugins`	Collects Oracle Enterprise Manager Plug-ins (EMPLUGINS) logs.
`-em`	Collects Oracle Enterprise Manager (EM) logs.
`-acfs`	Oracle Advanced Cluster File System (Oracle ACFS).
`-install`	Collects Oracle Installation related files.
`-cfgtools`	Collects configuration tools logs.
`-os`	Collects operating system files such as `/var/log/messages`.
`-ashhtml`	Collects Generate Active Session History (ASH) HTML report.
`-ashtext`	Collects Generate Active Session History (ASH) text report.
`-awrhtml`	Collects Automatic Workload Repository (AWR) HTML logs.
`-awrtext`	Collects Automatic Workload Repository (AWR) text report.

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.6 Collecting from Specific Directories

Oracle Trace File Analyzer discovers all Oracle diagnostics and collects relevant files based on the type and last time updated.

If you want to collect other files, then you can specify extra directories. Oracle Trace File Analyzer collects only the files updated in the relevant time range (one hour by default).

You can configure collection of all files irrespective of the time last updated. Configure on a directory by directory basis using the `-collectall` option.

To collect from specific directories:

1. To include all files updated in the last one hour:

   ```
   tfactl diagcollect -collectdir dir1,dir2,...dirn
   ```

   For example:

   To trim and collect all Oracle Clusterware files updated in the last one hour 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 configure Oracle Trace File Analyzer to collect all files from a directory, first configure it with the `-collectall` option:

   ```
   $ tfactl add dir -collectall
   ```

   or

   ```
   tfactl modify dir -collectall
   ```

   Start a diagnostic collection using the `-collectalldirs` option:

   ```
   $ tfactl diagcollect -collectalldirs
   ```

   > **Note:**
   >
   > If the `-collectalldirs` option is not used normal, then the file type, name, and time range restrictions are applied.

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.7 Changing the Collection Name

Oracle Trace File Analyzer zips collections and puts the zip files in the repository directory using the following naming format:

```
repository/collection_date_time/node_all/node.tfa_date_time.zip
```

You must only change the name of the zipped files using the following options. Manually changing the file name prevents you from using collections with various Oracle Support self-service tools.

To change the collection name:

1. To use your own naming to organize collections:

   ```
   -tag tagname
   ```

   The files are collected into *tagname* directory inside the repository.

2. To rename the `zip` file:

   ```
   -z zip name
   ```

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.8 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 files around the relevant time

To prevent copying zip files and trimming files:

1. To prevent copying the zip file back to the initiating node:

   ```
   -nocopy
   ```

   For example:

   ```
   $ tfactl diagcollect -last 1d -nocopy
   ```

2. To avoid trimming files:

   ```
   -notrim
   ```

   For example:

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

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.9 Performing Silent Collection

- To initiate a silent collection:

  ```
  -silent
  ```

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

  For example:

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

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.10 Collecting Core Files

- To collect core files:

  ```
  -cores
  ```

  For example:

  ```
  $ tfactl diagcollect -last 1d -cores
  ```

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.

8.9.11 Collecting Incident Packaging Service (IPS) Packages

Incident Packaging Service packages details of problems stored by Oracle Database in ADR for later diagnosis.

Oracle Trace File Analyzer runs IPS to query and collect these packages.

Syntax

```
tfactl ips option
```

Table 8-9 tfactl ips Command Parameters

Command	Description
`tfactl ips`	Runs the IPS.
`tfactl ips show incidents`	Shows all IPS incidents.
`tfactl ips show problems`	Shows all IPS problems.
`tfactl ips show package`	Shows all IPS Packages.
`tfactl diagcollect -ips -h`	Shows all available `diagcollect` IPS options.
`tfactl diagcollect -ips`	Performs an IPS collection following prompts. You can use all the standard `diagcollect` options to limit the scope of IPS collection.
`tfactl diagcollect -ips -adrbasepath` *`adr_base`* `-adrhomepath` *`adr_home`*	Performs an IPS collection in silent mode.
`tfactl diagcollect -ips -incident` *`incident_id`*	Collects ADR details about a specific incident id.
`tfactl diagcollect -ips -problem` *`problem_id`*	Collect ADR details about a specific problem id.

You can change the contents of the IPS package. Use the following options:

1. Start the collection.
2. Suspend the collection using the `-manageips` option.

For example:

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

3. Find the suspended collection using the `print suspendedips` option.

   For example:

   ```
   $ tfactl print suspendedips
   ```

4. Manipulate the package.
5. Resume the collection using the `-resumeips` option.

   For example:

   ```
   $ tfactl diagcollect -resumeips collection_id
   ```

Related Topics

- tfactl ips
  Use the `tfactl ips` command to collect Automatic Diagnostic Repository diagnostic data.

9 Using REST Service

Learn to configure REST service, and use REST service APIs.

- Configuring REST Service Using ORDS
  Oracle Trace File Analyzer includes REST support allowing invocation and query over HTTPS.
- Configuring REST Service Using Apache Tomcat
  The Oracle Trace File Analyzer installation includes a Web Application Resource (WAR) file to enable the REST service via Apache Tomcat.
- REST Service print API
  Learn to use the REST Service `print` API
- REST Service diagcollect API
  Learn to use the REST Service `diagcollect` API.
- REST Service download API
  Learn to use the REST Service `download` API.
- REST Service run API
  Learn to use REST Service `run` API.
- REST Service user API
  Learn to use REST Service `user` API. Log in as `tfaadmin` user to access these REST endpoints.

9.1 Configuring REST Service Using ORDS

Oracle Trace File Analyzer includes REST support allowing invocation and query over HTTPS.

tfactl rest Command-Line Options to Configure REST Service

Syntax

To facilitate this REST support Oracle REST Data Services (ORDS) is included within the installation.

```
tfactl rest [-status|-start|-stop|] [-dir dir] [-port port] [-user user] [-
debug [-level]]
```

> **Note:**
>
> You can run the REST command only as `root` user.

Parameters

Table 9-1 REST Command Parameters

Parameter	Description
`-status`	Prints the current status.
`-start`	Starts Oracle Trace File Analyzer REST services if not already running.
`-stop`	Stops Oracle Trace File Analyzer REST services if running.
`-dir` *`dir`*	Specify the directory to use to store the Oracle Trace File Analyzer REST configuration details. Defaults to the users home directory.
`-port` *`port`*	Specify the port to run ORDS on. Defaults to 9090.
`-user` *`user`*	Specify the user to run ORDS. Defaults to the GRID owner.
`-debug`	Enables debug.
`-level`	The level of debug to use, where available levels are: • 1 – FATAL • 2 – ERROR • 3 – WARNING • 4 – INFO (default) • 5 – DEBUG • 6 – TRACE

Once ORDS is running, you can invoke REST using the following APIs using requests of the form:

```
https://host:port/ords/api
```

For example:

```
https://host:port/ords/tfactl/print/status
```

REST Authentication

Oracle Trace File Analyzer REST uses first-party cookie-based authentication (basic authentication).

The Oracle Trace File Analyzer REST application is able to authenticate and authorize itself to the RESTful API using the same cookie session that the web application is using. The first party application has full access to the RESTful API.

During start-up Oracle Trace File Analyzer prompts you for the password for the `tfaadmin` and `tfarest` users.

- Use `tfarest` user for REST calls

- Use tfaadmin for making REST calls and to manage the REST service, for example, changing the logging level

```
# tfactl rest -start

Configuring TFA REST Services using ORDS :

This might take couple of minutes. Please be patient.

Adding Dependency Jars to ORDS

Adding users to ORDS :

Enter a password for user tfaadmin:
Confirm password for user tfaadmin:

Enter a password for user tfarest:
Confirm password for user tfarest:

Starting TFA REST Services

Successfully started TFA REST Services [PID : 32650]

URL : https://myserver:9090/ords/tfactl/print/status
```

Access the web service from a browser using the following URL:

```
https://host_name:9090/ords/tfactl/print/status
```

You are presented with a 401 message, which includes a **sign in** link. Click the link, sign in with tfarest credentials you just created, and you will be directed to REST output.

Alternatively, you can also specify the credentials in a curl command.

```
# curl -k --user tfarest:mypassword https://myserver:9090/ords/tfactl/print/
status
[ {
  "status" : "CheckOK",
  "hostname" : "myserver",
  "pid" : 2430,
  "port" : 5000,
  "version" : "latest-version",
  "buildId" : "latest-build-ID",
  "inventoryStatus" : "COMPLETE"
} ]
```

9.2 Configuring REST Service Using Apache Tomcat

The Oracle Trace File Analyzer installation includes a Web Application Resource (WAR) file to enable the REST service via Apache Tomcat.

To enable the REST service using Apache Tomcat:

1. Deploy the WAR file located at TFA_HOME/common/jlib/tfa.war to your Tomcat server.

2. Change the `tfaadmin` user password.

   ```
   curl -k --user tfaadmin:tfaadmin -X POST "https://host/tfa/tfactl/user/
   update" '{ "password" : "some_new_password" }'
   ```

3. Change the `tfarest` user password.

   ```
   curl -k --user tfarest:tfarest -X POST "https://host/tfa/tfactl/user/
   update" '{ "password" : "some_new_password" }'
   ```

4. Add the Tomcat user to the Oracle Trace File Analyzer access list.

   ```
   tfactl access add -user tomcat_user
   ```

9.3 REST Service print API

Learn to use the REST Service `print` API

- status
  Use GET requests to print the statuses of all hosts.
- hosts
  Use GET requests to print the list of hosts.
- actions
  Use GET requests to print the list of actions performed on all hosts.
- repository
  Use GET requests to print the repository details of all hosts.
- collections
  Use GET requests to print the details of all collections, or a specific collection.
- config
  Use GET requests to print the configuration details of all hosts.
- protocols
  Use GET requests to print the details of protocols of all hosts.
- directories
  Use GET requests to print the details of directories of all hosts.

9.3.1 status

Use GET requests to print the statuses of all hosts.

Syntax

```
/tfactl/print/status
```

Example 9-1   print

```
[ {
  "status" : "CheckOK",
  "hostname" : "myhost",
  "pid" : 73637,
  "port" : 9090,
```

```
  "version" : "latest-version",
  "buildId" : "latest-build-ID",
  "inventoryStatus" : "COMPLETE"
} ]
```

9.3.2 hosts

Use GET requests to print the list of hosts.

Syntax

```
/tfactl/print/hosts
```

Example 9-2   hosts

```
[ {
  "hostname" : "myhost"
} ]
```

9.3.3 actions

Use GET requests to print the list of actions performed on all hosts.

Syntax

```
/tfactl/print/actions
```

Example 9-3   actions

```
[ {
  "actionName" : "Run inventory",
  "hostname" : "Requested in all nodes",
  "client" : "tfactl",
  "startTime" : "Jan 09 07:50:26 PST",
  "endTime" : "Jan 09 07:50:29 PST",
  "status" : "COMPLETE",
  "comments" : null
} ]
```

9.3.4 repository

Use GET requests to print the repository details of all hosts.

Syntax

```
/tfactl/print/repository
```

Example 9-4   repository

```
[ {
  "hostname" : "myhost",
  "directory" : "/scratch/smith/view_storage/smith_tfa_latest/oracle/log/tfa/
repository",
  "status" : "OPEN",
  "maxSizeMB" : 10240,
  "currentSizeMB" : 13,
  "freeSpaceMB" : 10227
} ]
```

9.3.5 collections

Use GET requests to print the details of all collections, or a specific collection.

Syntax

```
/tfactl/print/collections
/tfactl/print/collections/{collectionid}
```

Example 9-5 collections

```
[ {
  "id" : "20171010115528myhost",
  "type" : "Manual Collection",
  "requestUser" : "smith",
  "nodeList" : "[]",
  "masterHost" : "myhost",
  "startTime" : "Mon Oct 09 23:55:32 PDT 2017",
  "endTime" : "Tue Oct 10 11:55:32 PDT 2017",
  "tag" : "/scratch/smith/view_storage/smith_tfa_latest/oracle/log/tfa/
repository/tfa_11",
  "zipFileName" : "myhost.tfa_Tue_Oct_10_11_55_28_PDT_2017.zip",
  "componentList" : "[emagent, crsclient, oms, dbwlm,emplugins, cfgtools,
afd, wls]",
  "zipFileSize" : 3055,
  "collectionTime" : 16,
  "events" : null
}]


[{
  "id" : "20171011044112myhost",
  "type" : "Manual Collection",
  "requestUser" : "smith",
  "nodeList" : "[]",
  "masterHost" : "myhost",
  "startTime" : "null",
  "endTime" : "Wed Oct 11 04:41:14 PDT 2017",
  "tag" : "/scratch/smith/view_storage/smith_tfa_latest/oracle/log/tfa/
repository/TFA_T1",
  "zipFileName" : "myhost.TFA_T1.zip",
  "componentList" : "[]",
  "zipFileSize" : 0,
  "collectionTime" : 0,
  "events" : null
}]
```

9.3.6 config

Use GET requests to print the configuration details of all hosts.

Syntax

```
/tfactl/print/config
```

Example 9-6   config

```
[ {
  "hostname" : "myhost",
  "tfaVersion" : "latest-version",
  "javaVersion" : "latest-version",
  "inventoryTraceLevel" : 1,
  "collectionTraceLevel" : 1,
  "scanTraceLevel" : 1,
  "otherTraceLevel" : 3,
  "currentSizeMB" : 13,
  "maxSizeMB" : 10240,
  "maxLogSize" : 50,
  "maxLogCount" : 10,
  "maxCoreFileSize" : 50,
  "maxCoreCollectionSize" : 500,
  "minSpaceForRTScan" : 500,
  "diskUsageMoninterInterval" : 60,
  "manageLogsAutoPurgeInterval" : 60,
  "manageLogsAutoPurgePolicyAge" : "30d",
  "minFileAgeToPurge" : 12,
  "language" : "en",
  "encoding" : "UTF-8",
  "country" : "US",
  "alertLogLevel" : "ALL",
  "userLogLevel" : "ALL",
  "baseLogPath" : "ERROR",
  "tfaIpsPoolSize" : 5,
  "autoPurge" : true,
  "publicIp" : false,
  "fireZipsInRT" : true,
  "rtscan" : true,
  "diskUsageMonOn" : true,
  "manageLogsAutoPurgeOn" : false,
  "trimmingOn" : true
} ]
```

9.3.7 protocols

Use GET requests to print the details of protocols of all hosts.

Syntax

```
/tfactl/print/protocols
```

Example 9-7 protocols

```
{
  "hostname" : "myhost",
  "available" : [ "TLSv1.2" ],
  "restricted" : [ "SSLv3", "SSLv2Hello", "TLSv1", "TLSv1.1" ]}
```

9.3.8 directories

Use GET requests to print the details of directories of all hosts.

Syntax

```
/tfactl/print/directories
```

Example 9-8 directories

```
[ {
  "hostname" : "myhost",
  "directory" : "/oem/app/oracle/product/emagent/agent_inst/install/logs",
  "components" : [ "EMPLUGINS" ],
  "permission" : "public",
  "owner" : "root",
  "collectionPolicy" : "exclusions",
  "collectAll" : false
}, {
  "hostname" : "myhost",
  "directory" : "/oem/app/oracle/product/emagent/agent_inst/sysman/log",
  "components" : [ "EMAGENT" ],
  "permission" : "public",
  "owner" : "root",
  "collectionPolicy" : "exclusions",
  "collectAll" : false
} ]
```

9.4 REST Service diagcollect API

Learn to use the REST Service `diagcollect` API.

- diagcollect
  Use POST requests to view collection details.

9.4.1 diagcollect

Use POST requests to view collection details.

Syntax

```
/tfactl/diagcollect
```

Returns

Oracle Trace File Analyzer default collection for last one hour for all components.

Or, Oracle Trace File Analyzer collection per JSON data as parameters specified.

Example 9-9 diagcollect–default collection

```
testuser: {
  "collectionId" : "20190401121115slc13lyb",
  "zipName" : "TFA_DEF_ZIP_20190401121115",
  "tagName" : "TFA_DEF_TAG_20190401121115",
  "message" : [ "Diagcollect request will be processed soon by TFA" ]
}
```

Example 9-10 diagcollect–JSON data as Parameters

Input:

```
[{
  "components": "-database -asm -tns -crs -acfs -install -cfgtools -os",
  "timePeriod": "-since n[d|h] | -last n[d|h] | -for date |
        -from date -to date",
  "tagName": "crs_crash_collection",
  "nodeList": "node1,node2",
  "options": "-nocopy | -notrim | -silent | -cores |
       -collectalldirs | -collectdir dir1,dir2..."
}]
```

Output:

```
[{
  "collectionId" : "20180111011121slc12ekf",
  "zipName" : "TFA_DEF_ZIP_20180111011121",
  "tagName" : "TFA_DEF_TAG_20180111011121"
}]
```

9.5 REST Service download API

Learn to use the REST Service `download` API.

- download
  Use GET requests to download collection ZIP file for a specific collection ID.

9.5.1 download

Use GET requests to download collection ZIP file for a specific collection ID.

Syntax

```
/tfactl/download/{collectionid}
```

Returns

Collection ZIP file for the collection ID specified.

Usage Notes

Specify the collection ID for which you want to download the collection ZIP file.

9.6 REST Service run API

Learn to use REST Service `run` API.

- alertsummary
  Use GET requests to run the `alertsummary` command.
- calog
  Use GET requests to run the `calog` command.
- changes
  Use GET requests to run the `changes` command.
- events
  Use GET requests to run the `events` command.
- history
  Use GET requests to run the `history` command.

9.6.1 alertsummary

Use GET requests to run the `alertsummary` command.

Syntax

```
/tfactl/run/alertsummary
```

Returns

Runs the `alertsummary` command and returns the alert summary.

Example 9-11   alertsummary

```
[ {
  "line" : "Output from host : myserver"
}, {
  "line" : "-----------------------------"
}, {
  "line" : "Reading /scratch/app/oradb/diag/rdbms/apxcmupg/apxcmupg_2/trace/
alert_apxcmupg_2.log"
}, {
  "line" : "+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
+-+-+-"
}, {
  "line" :
"----------------------------------------------------------------------"
}, {
  "line" : "Oct 20 08:20:43 Database started"
}, {
  "line" :
"-------------------------------------------------------------------------"
}, {
  "line" : "Nov 05 20:27:50 Database started"
},
....
....
```

9.6.2 calog

Use GET requests to run the `calog` command.

Syntax

```
/tfactl/run/calog
```

9.6.3 changes

Use GET requests to run the `changes` command.

Syntax

```
/tfactl/run/changes
```

Example 9-12 changes

```
[ {
  "line" : "Output from host : myserver"
}, {
  "line" : "------------------------------"
}, {
  "line" : "[Jul/25/2018 03:45:15.325]: Parameter: fs.aio-nr: Value: 276224
=> 277760"
}, {
  "line" : "[Jul/25/2018 03:45:15.325]: Parameter:
kernel.random.entropy_avail: Value: 192 => 152"
}, {
  "line" : "[Jul/25/2018 03:45:15.325]: Parameter: kernel.random.uuid:
Value: 5eac06d7-560a-466d-a035-efe836fe0b57 => 3f329d9c-25d3-4057-
ab00-17d031645490"
}, {
  "line" : "[Jul/25/2018 15:46:15.325]: Parameter: fs.aio-nr: Value: 277760
=> 279296"
},
....
....
```

9.6.4 events

Use GET requests to run the `events` command.

Syntax

```
/tfactl/run/events
```

Example 9-13   events

```
[ {
  "line" : "Output from host : myserver"
}, {
  "line" : "------------------------------"
}, {
  "line" : "Event Summary:"
}, {
  "line" : "INFO    :0"
}, {
  "line" : "ERROR   :0"
}, {
  "line" : "WARNING :0"
}, {
  "line" : "Event Timeline:"
}, {
  "line" : "No Events Found"
} ]
```

9.6.5 history

Use GET requests to run the `history` command.

Syntax

```
/tfactl/run/history
```

9.7 REST Service user API

Learn to use REST Service `user` API. Log in as `tfaadmin` user to access these REST endpoints.

- add
  Use POST requests to add users to Oracle Trace File Analyzer REST Services.
- delete
  Use POST requests to delete an Oracle Trace File Analyzer REST Services user.
- update
  Use POST requests to update the password of an Oracle Trace File Analyzer REST Services user.

9.7.1 add

Use POST requests to add users to Oracle Trace File Analyzer REST Services.

Syntax

```
/tfactl/user/add
```

Example 9-14 add

Input:

```
{
  "userName" : "test",
  "password" : "test"
}
```

Output:

```
{
  "status": "SUCCESS",
  "message": "Successfully added test to TFA REST Services"
}
```

9.7.2 delete

Use POST requests to delete an Oracle Trace File Analyzer REST Services user.

Syntax

```
/tfactl/user/delete
```

Example 9-15 delete

Input:

```
{
  "userName" : "test"
}
```

Output:

```
{
  "status": "SUCCESS",
  "message": "Successfully removed test from TFA REST Services"
}
```

9.7.3 update

Use POST requests to update the password of an Oracle Trace File Analyzer REST Services user.

Syntax

```
/tfactl/user/update
```

Example 9-16 update

Input:

```
{
  "password" : "test"
}
```

Output:

```
{
  "status": "SUCCESS",
  "message": "Successfully updated users's profile in TFA"
}
```

10 Managing and Configuring Oracle Trace File Analyzer

This section helps you manage Oracle Trace File Analyzer daemon, diagnostic collections, and the collection repository.

- Querying Oracle Trace File Analyzer Status and Configuration
  Use the `print` command to query the status or configuration.
- Managing the Oracle Trace File Analyzer Daemon
  Oracle Trace File Analyzer runs from `init` on UNIX systems or `init/upstart/systemd` on Linux, or Microsoft Windows uses a Windows Service so that Oracle Trace File Analyzer starts automatically whenever a node starts.
- Managing the Repository
  Oracle Trace File Analyzer stores all diagnostic collections in the repository.
- Managing Collections
  Manage directories configured in Oracle Trace File Analyzer and diagnostic collections.
- Configuring the Host
  You must have `root` or `sudo` access to `tfactl` to add hosts to Oracle Trace File Analyzer configuration.
- Configuring the Ports
  The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005.
- Configuring SSL and SSL Certificates
  View and restrict SSL/TLS protocols. Configure Oracle Trace File Analyzer to use self-signed or CA-signed certificates.
- Configuring Email Notification Details
  Configure Oracle Trace File Analyzer to send an email to the registered email address after an automatic collection completes.
- Managing the Index
  Oracle Trace File Analyzer uses multiple indexes to store diagnostic data.

10.1 Querying Oracle Trace File Analyzer Status and Configuration

Use the `print` command to query the status or configuration.

Table 10-1   Configuration Listing and Descriptions

Configuration Listing	Default Value	Description
Automatic diagnostic collection	ON	Triggers a collection if a significant problem occurs. Possible values: • ON • OFF
Trimming of files during diagnostic collection	ON	Trims the log files to only entries within the time range of the collection. Possible values: • ON • OFF
Repository maximum size in MB	Smaller of either 10GB or 50% of free space in the file system.	The largest size the repository can be.
Trace Level	INFO	Increases the level of verbosity. Possible values: • FATAL • ERROR • WARNING • INFO • DEBUG • TRACE A value of INFO results in the least amount of trace. A value of TRACE results in the most amount of trace. Oracle recommends changing the trace level value only at the request of Oracle Support.
Automatic Purging	ON	Purges collections when: Free space in the repository falls below 1 GB. Or Before closing the repository. Purging removes collections from largest size through to smallest. Purging continues until the repository has enough space to open.
Minimum Age of Collections to Purge (Hours)	12	The least number of hours to keep a collection, after which it is eligible for purging.
Minimum Space free to enable Alert Log Scan (MB)	500	Suspends log scanning if free space in the `tfa_home` falls below this value.

Related Topics

- tfactl print
  Use the `tfactl print` command to print information from the Berkeley DB (BDB).

10.2 Managing the Oracle Trace File Analyzer Daemon

Oracle Trace File Analyzer runs from `init` on UNIX systems or `init/upstart/systemd` on Linux, or Microsoft Windows uses a Windows Service so that Oracle Trace File Analyzer starts automatically whenever a node starts.

To manage Oracle Trace File Analyzer daemon:

The `init` control file `/etc/init.d/init.tfa` is platform dependant.

1. To start or stop Oracle Trace File Analyzer manually:
   - `tfactl start`: Starts the Oracle Trace File Analyzer daemon
   - `tfactl stop`: Stops the Oracle Trace File Analyzer daemon

   If the Oracle Trace File Analyzer daemon fails, then the operating system restarts the daemon automatically.
2. To enable or disable automatic restarting of the Oracle Trace File Analyzer daemon:
   - `tfactl disable`: Disables automatic restarting of the Oracle Trace File Analyzer daemon.
   - `tfactl enable`: Enables automatic restarting of the Oracle Trace File Analyzer daemon.

10.3 Managing the Repository

Oracle Trace File Analyzer stores all diagnostic collections in the repository.

The repository size is the maximum space Oracle Trace File Analyzer is able to use on disk to store collections.

- Purging the Repository Automatically
- Purging the Repository Manually

10.3.1 Purging the Repository Automatically

Oracle Trace File Analyzer closes the repository, if:

- Free space in `TFA_HOME` is less than 100 MB, also stops indexing
- Free space in `ORACLE_BASE` is less than 100 MB, also stops indexing
- Free space in the repository is less than 1 GB
- Current size of the repository is greater than the repository max size (`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
  Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

10.3.2 Purging the Repository Manually

To purge the repository manually:

1. To view the status of the Oracle Trace File Analyzer repository:

   ```
   tfactl print repository
   ```

2. To view statistics about collections:

   ```
   tfactl print collections
   ```

3. To manually purge collections that are older than a specific time:

   ```
   tfactl purge -older number[h|d] [-force]
   ```

Related Topics

- tfactl purge
  Use the `tfactl purge` command to delete diagnostic collections from the Oracle Trace File Analyzer repository that are older than a specific time.
- tfactl print
  Use the `tfactl print` command to print information from the Berkeley DB (BDB).

10.4 Managing Collections

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

- Including Directories
  Add directories to the Oracle Trace File Analyzer configuration to include the directories in diagnostic collections.
- Managing the Size of Collections
  Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and `diagcollect -cores` to include core files.
- Temporarily Restrict Automatic Diagnostic Collections for Specific Events
  Use the `tfactl blackout` command to suppress automatic diagnostic collections.

10.4.1 Including Directories

Add directories to the Oracle Trace File Analyzer configuration to include the directories in diagnostic collections.

Oracle Trace File Analyzer then stores diagnostic collection metadata about the:

- Directory
- Subdirectories
- Files in the directory and all sub directories

All Oracle Trace File Analyzer users can add directories they have read access to.

To manage directories:

1. To view the current directories configured in Oracle Trace File Analyzer

   ```
   tfactl print directories [ -node all | local | n1,n2,... ]
   [ -comp component_name1,component_name2,.. ]
   [ -policy  exclusions | noexclusions ]
   [ -permission public | private ]
   ```

2. To add directories:

   ```
   tfactl directory add dir
   [ -public ]
   [ -exclusions | -noexclusions | -collectall ]
   [ -node all | n1,n2,... ]
   ```

3. To remove a directory from being collected:

   ```
   tfactl directory remove dir [ -node all | n1,n2,... ]
   ```

Related Topics

- tfactl directory
  Use the `tfactl directory` command to add a directory to, or remove a directory from the list of directories to analyze their trace or log files.
- tfactl print
  Use the `tfactl print` command to print information from the Berkeley DB (BDB).

10.4.2 Managing the Size of Collections

Use the Oracle Trace File Analyzer configuration options `trimfiles`, `maxcorefilesize`, `maxcorecollectionsize`, and `diagcollect -cores` to include core files.

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 collect core files with diagnostic collections:

   ```
   tfactl diagcollect -cores
   ```

Related Topics

- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.
- tfactl set
  Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.

10.4.3 Temporarily Restrict Automatic Diagnostic Collections for Specific Events

Use the `tfactl blackout` command to suppress automatic diagnostic collections.

If you set blackout for a target, then Oracle Trace File Analyzer stops automatic diagnostic collections if it finds events in the alert logs for that target while scanning.

You can also restrict automatic diagnostic collection at a granular level, for example, only for `ORA-00600` or even only `ORA-00600` with specific arguments.

```
tfactl blackout add -targettype database -target mydb -event "ORA-00600"
```

Event "ORA-00600" is blacked out until Wed Feb 20 00:20:34 PST 2019 on targettype : database, target : mydb

You can also blackout a resource that does not exist yet. For example, if you want to create a database and you do not want to care about the status until the provisioning is completed, then do as follows:

1. Blackout the database you are about to create
2. Create the database
3. Remove the blackout

10.5 Configuring the Host

You must have `root` or `sudo` access to `tfactl` to add hosts to Oracle Trace File Analyzer configuration.

To add, remove, and replace SSL certificates:

1. To view the list of current hosts in the Oracle Trace File Analyzer configuration:

   ```
   tfactl print hosts
   ```

2. To add a host to the Oracle Trace File Analyzer configuration for the first time:
   a. If necessary, install and start Oracle Trace File Analyzer on the new host.
   b. From the existing host, synchronize authentication certificates for all hosts by running:

      ```
      tfactl syncnodes
      ```

      If needed, then Oracle Trace File Analyzer displays the current node list it is aware of and prompts you to update this node list.
   c. Select **Y**, and then enter the name of the new host.

Oracle Trace File Analyzer contacts Oracle Trace File Analyzer on the new host to synchronize certificates and add each other to their respective hosts lists.

3. To remove a host:

   ```
   tfactl host remove host
   ```

4. To add a host and the certificates that are already synchronized:

   ```
   tfactl host add host
   ```

   Oracle Trace File Analyzer generates self-signed SSL certificates during installation. Replace those certificates with one of the following:

   - Personal self-signed certificate
   - CA-signed certificate

10.6 Configuring the Ports

The Oracle Trace File Analyzer daemons in a cluster communicate securely over ports 5000 to 5005.

If the port range is not available on your system, then replace it with the ports available on your system.

To change the ports:

1. To set the primary port use the `tfactl set port` command:

   ```
   tfactl set port=port_1
   ```

   Or, specify a comma-delimited list of sequentially numbered ports to use. You can specify a maximum of five ports.

   ```
   tfactl set port=port_1,port_2,port_3,port_4,port_5
   ```

2. Restart Oracle Trace File Analyzer on all nodes:

   ```
   tfactl restart
   ```

10.7 Configuring SSL and SSL Certificates

View and restrict SSL/TLS protocols. Configure Oracle Trace File Analyzer to use self-signed or CA-signed certificates.

- Configuring SSL/TLS Protocols
  The Oracle Trace File Analyzer daemons in a cluster communicate securely using the SSL/TLS protocols.
- Configuring Self-Signed Certificates
  Use `Java keytool` to replace self-signed SSL certificates with personal self-signed certificates.

- Configuring CA-Signed Certificates
  Use `Java keytool` and `openssl` to replace self-signed SSL certificates with the Certificate Authority (CA) signed certificates.
- Configuring SSL Cipher Suite
  The cipher suite is a set of cryptographic algorithms used by the TLS/SSL protocols to create keys and encrypt data.

10.7.1 Configuring SSL/TLS Protocols

The Oracle Trace File Analyzer daemons in a cluster communicate securely using the SSL/TLS protocols.

The SSL protocols available for use by Oracle Trace File Analyzer are:

- `TLSv1.2`
- `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
   ```

10.7.2 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. Restrict the permissions on the keystores to `root read-only`.

   ```
   $ chmod 400 myclient.jks myserver.jks
   ```

8. Copy the keystores (`jks` files) to each node.

9. Configure Oracle Trace File Analyzer to use the new certificates:

   ```
   $ tfactl set sslconfig
   ```

10. Restart the Oracle Trace File Analyzer process to start using new certificates:

    ```
    $ tfactl restart
    ```

10.7.3 Configuring CA-Signed Certificates

Use `Java keytool` and `openssl` to replace self-signed SSL certificates with the Certificate Authority (CA) signed certificates.

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

1. Create a private key for the server request:

   ```
   $ openssl genrsa -aes256 -out myserver.key 2048
   ```

2. Create a private key for the client request:

   ```
   $ openssl genrsa -aes256 -out myclient.key 2048
   ```

3. Create a Certificate Signing Request (CSR) for the server:

   ```
   $ openssl req -key myserver.key -new -sha256 -out myserver.csr
   ```

4. Create a Certificate Signing Request (CSR) for the client:

   ```
   $ openssl req -key myclient.key -new -sha256 -out myclient.csr
   ```

5. Send the resulting CSR for the client and the server to the relevant signing authority.

   The signing authority sends back the signed certificates:

   - `myserver.cert`
   - `myclient.cert`
   - CA root certificate

6. Convert the certificates to JKS format for the server and the client:

   ```
   $ openssl pkcs12 -export -out serverCert.pkcs12 -in myserver.cert -inkey
   myserver.key
   ```

   ```
   $ keytool -v -importkeystore -srckeystore serverCert.pkcs12 -srcstoretype
   PKCS12 -destkeystore myserver.jks -deststoretype JKS
   ```

   ```
   $ openssl pkcs12 -export -out clientCert.pkcs12 -in myclient.cert -inkey
   myclient.key
   ```

   ```
   $ keytool -v -importkeystore -srckeystore clientCert.pkcs12 -srcstoretype
   PKCS12 -destkeystore myclient.jks -deststoretype JKS
   ```

7. Import the server public key into to the client `jks` file:

   ```
   $ keytool -import -v -alias server-ca -file myserver.cert -keystore
   myclient.jks
   ```

8. Import the client public key to the server `jks` file:

   ```
   $ keytool -import -v -alias client-ca -file myclient.cert -keystore
   myserver.jks
   ```

9. Import the CA root certificate from the signing authority into the Oracle Trace File Analyzer server certificate:

   ```
   $ keytool -importcert -trustcacerts -alias inter -file caroot.cert -
   keystore myserver.jks
   ```

10. Restrict the permissions on the keystores to `root read-only`:

    ```
    $ chmod 400 myclient.jks myserver.jks
    ```

11. Copy the keystores (`jks` files) to each node.

12. Configure Oracle Trace File Analyzer to use the new certificates:

    ```
    $ tfactl set sslconfig
    ```

13. Restart the Oracle Trace File Analyzer process to start using the new certificates.

    ```
    $ tfactl stop
    $ tfactl start
    ```

10.7.4 Configuring SSL Cipher Suite

The cipher suite is a set of cryptographic algorithms used by the TLS/SSL protocols to create keys and encrypt data.

Oracle Trace File Analyzer supports any of the cipher suites used by JRE 1.8.

The default cipher suite used is `TLS_RSA_WITH_AES_128_CBC_SHA256`.

- You can change the cipher suite with the command:

  ```
  tfactl set ciphersuite=cipher_suite
  ```

  For example:

  ```
  tfactl set ciphersuite=TLS_RSA_WITH_AES_128_GCM_SHA256
  ```

  For a list of JRE cipher suites, see:
  https://docs.oracle.com/javase/8/docs/technotes/guides/security/SunProviders.html#SunJSSEProvider

10.8 Configuring Email Notification Details

Configure Oracle Trace File Analyzer to send an email to the registered email address after an automatic collection completes.

To send emails, configure the system on which Oracle Trace Analyzer is running. You must configure notification with a user email address to enable it to work.

To configure email notification details:

1. To set the notification email to use for a specific ORACLE_HOME, include the operating system owner in the command:

   ```
   tfactl set notificationAddress=os_user:email
   ```

   For example:

   ```
   tfactl set notificationAddress=oracle:some.body@example.com
   ```

2. To set the notification email to use for any ORACLE_HOME:

   ```
   tfactl set notificationAddress=email
   ```

   For example:

   ```
   tfactl set notificationAddress=another.body@example.com
   ```

3. Configure the SMTP server using `tfactl set smtp`.

   Set the SMTP parameters when prompted.

   Table 10-2 tfactl diagnosetfa Command Parameters

Parameter	Description
`smtp.host`	Specify the SMTP server host name.
`smtp.port`	Specify the SMTP server port.
`smtp.user`	Specify the SMTP user.
`smtp.password`	Specify password for the SMTP user.
`smtp.auth`	Set the Authentication flag to true or false.
`smtp.ssl`	Set the SSL flag to true or false.
`smtp.from`	Specify the from mail ID.
`smtp.to`	Specify the comma-delimited list of recipient mail IDs.
`smtp.cc`	Specify the comma-delimited list of CC mail IDs.
`smtp.bcc`	Specify the comma-delimited list of BCC mail IDs.
`smtp.debug`	Set the Debug flag to true or false.

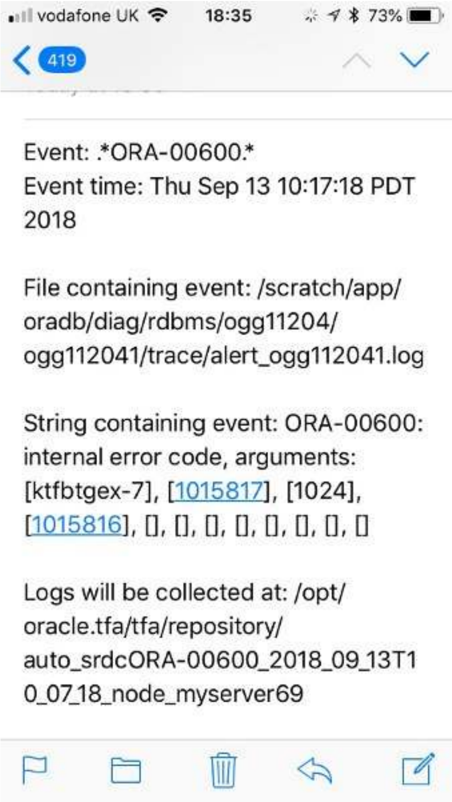
Note:

You can view current SMTP configuration details using `tfactl print smtp`.

4. Verify SMTP configuration by sending a test email using `tfactl sendmail` *`email_address`*.

   When Oracle Trace File Analyzer detects a significant error has occurred it will send an email notification as follows:

Figure 10-1 Email Notification



Event: .*ORA-00600.*
Event time: Thu Sep 13 10:17:18 PDT 2018

File containing event: /scratch/app/oradb/diag/rdbms/ogg11204/ogg112041/trace/alert_ogg112041.log

String containing event: ORA-00600: internal error code, arguments: [ktfbtgex-7], [1015817], [1024], [1015816], [], [], [], [], [], [], [], []

Logs will be collected at: /opt/oracle.tfa/tfa/repository/auto_srdcORA-00600_2018_09_13T10_07_18_node_myserver69

5. Do the following after receiving the notification email:
   a. To find the root cause, inspect the referenced collection details.
   b. If you can fix the issue, then resolve the underlying cause of the problem.
   c. If you do not know the root cause of the problem, then log an SR with Oracle Support, and upload the collection details.

10.9 Managing the Index

Oracle Trace File Analyzer uses multiple indexes to store diagnostic data.

The DBA tools and diagnostic collections can use either an index (default), or the shipped Berkeley DB (BDB).

Using the index results in lower CPU usage and faster average execution times for diagnostic collections and the running of DBA tools such as `ls`, `grep`, `tail`, `vi`, and so on. However, using the index requires more ongoing resource consumption than the Berkeley DB (BDB).

If you do not use the DBA tools and are prepared to wait longer for collections to complete, you can disabled this indexing by running:

```
tfactl set indexInventory=false
```

ISA telemetry data is stored in a Lucene index. Occasionally this index can get corrupted. If corruption is detected then by default the index will be dropped and recreated. This can result in the loss of some ISA telemetry data.

If you do not want to risk losing any ISA data you can change this behavior to restore, so the index is backed up and redo data is maintained.

1. (Default) To drop and recreate, use:

   ```
   tfactl set indexRecoveryMode=recreate
   ```

2. To backup, maintain redo data and restore the index, use:

   ```
   tfactl set indexRecoveryMode=restore
   ```

11 Managing Oracle Database and Oracle Grid Infrastructure Logs

This section enables you to manage Oracle Database and Oracle Grid Infrastructure diagnostic data and disk usage snapshots.

- Managing Automatic Diagnostic Repository Log and Trace Files
  Use the `managelogs` command to manage Automatic Diagnostic Repository log and trace files.
- Managing Disk Usage Snapshots
  Use `tfactl` commands to manage Oracle Trace File Analyzer disk usage snapshots.
- Purging Oracle Database and Oracle Grid Infrastructure Logs
  Use these `tfactl` commands to manage log file purge policy for Oracle Database and Oracle Grid Infrastructure logs.
- Securing Access to Diagnostic Collections
  Running `tfactl` commands is restricted to authorized users.

11.1 Managing Automatic Diagnostic Repository Log and Trace Files

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

The `-purge` command option removes files managed by Automatic Diagnostic Repository. This command clears files from “`ALERT`”, "`INCIDENT`", "`TRACE`", "`CDUMP`", "`HM`", "`UTSCDMP`", "`LOG`" under diagnostic destinations. The `-purge` command also provides details about the change in the file system space.

If the diagnostic destinations contain large numbers of files, then the command runs for a while. Check the removal of files in progress from the corresponding directories.

To remove files, you must have operating system privileges over the corresponding diagnostic destinations.

To manage Automatic Diagnostic Repository log and trace files:

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

   ```
   $ tfactl managelogs -older 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. To get an estimate of how many files are removed and how much space is freed, use the `–dryrun` option:

   For example:

   ```
   $ tfactl managelogs -purge -older 30d -dryrun
   ```

3. To remove files and clean disk space:

   For example:

   ```
   $ tfactl managelogs -purge -older 30d

   $ tfactl managelogs -purge -older 30d -gi

   $ tfactl managelogs -purge -older 30d -database
   ```

4. To view the space usage of individual diagnostic destinations:

   For example:

   ```
   $ tfactl managelogs -show usage

   $ tfactl managelogs -show usage -gi

   $ tfactl managelogs -show usage -database
   ```

Related Topics

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

11.2 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_install_dir*/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
   ```

11.3 Purging Oracle Database and Oracle Grid Infrastructure Logs

Use these `tfactl` commands to manage log file purge policy for Oracle Database and Oracle Grid Infrastructure logs.

Automatic purging is enabled by default on a Domain Service Cluster (DSC), and disabled by default elsewhere. When automatic purging is enabled, every 60 minutes, Oracle Trace File Analyzer automatically purges logs that are older than 30 days.

To purge Oracle Trace File Analyzer logs automatically:

1. To turn on or off automatic purging:

   ```
   $ tfactl set manageLogsAutoPurge=ON|OFF
   ```

2. To adjust the age of logs to purge:

   ```
   $ tfactl set manageLogsAutoPurgePolicyAge=nd|h
   ```

3. To adjust the frequency of purging:

   ```
   $ tfactl set manageLogsAutoPurgeInterval=minutes
   ```

11.4 Securing Access to Diagnostic Collections

Running `tfactl` commands is restricted to authorized users.

`tfactl` provides a command-line interface and shell to do the following:

- Run diagnostics and collect all relevant log data from a time of your choosing
- Trim log files to collect 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 `tfactl` commands.

By default, the following users are authorized to access a subset of `tfactl` commands:

- Oracle Grid Infrastructure home owner
- Oracle Database home owners

User access is applicable only if Oracle Trace File Analyzer is installed as `root` on Linux and UNIX. User access is not applicable if Oracle Trace File Analyzer is installed as non-root, or on Microsoft Windows.

To provision user access to tfactl:

- To list the users who have access to `tfactl`:

  ```
  tfactl access lsusers
  ```

- To add a user to access `tfactl`:

  ```
  tfactl access add –user user [-local]
  ```

  By default, access commands are applicable to cluster-wide unless you specify the `-local` command option to restrict them to local node.

- To remove a user from accessing `tfactl`:

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

- To remove all users from accessing `tfactl`:

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

- To reset user access to default:

  ```
  tfactl access reset
  ```

- To enable user access:

  ```
  tfactl access enable
  ```

- To disable user access:

  ```
  tfactl access disable
  ```

Part V
Appendixes

- Compliance Framework (Oracle ORAchk and Oracle EXAchk) Command-Line Options
  Review the list of commands that you can use to run compliance checks on Oracle Engineered and non-engineered systems.
- OCLUMON Command Reference
  Use the command-line tool to query the Cluster Health Monitor repository to display node-specific metrics for a specific time period.
- Managing the Cluster Resource Activity Log
  Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.
- chactl Command Reference
  The Oracle Cluster Health Advisor commands enable the Oracle Grid Infrastructure user to administer basic monitoring functionality on the targets.
- Oracle Autonomous Health Framework Command-Line and Shell Options
  TFACTL is the command-line interface for Oracle Trace File Analyzer.
- Behavior Changes, Deprecated and Desupported Features
  Review information about changes, deprecations, and desupports.

A

Compliance Framework (Oracle ORAchk and Oracle EXAchk) Command-Line Options

Review the list of commands that you can use to run compliance checks on Oracle Engineered and non-engineered systems.

- Compliance Framework (Oracle ORAchk and Oracle EXAchk) Command-Line Options
  Review the list of Compliance Framework (Oracle ORAchk and Oracle EXAchk) command-line options.
- Running Generic Compliance Framework (Oracle ORAchk and Oracle EXAchk) Commands
  Review the list of generic Oracle ORAchk and Oracle EXAchk command options.
- Controlling the Scope of Checks
  Use the list of commands to control the scope of checks.
- Managing the Report Output
  Use the list of commands to manage compliance checks report output.
- Uploading Results to Database
  Use the list of commands to upload results to the database.
- Controlling the Behavior of the Daemon
  Use the list of commands to control the behavior of the daemon.
- Tracking File Attribute Differences
  Use the list of commands to track file attribute differences.
- Running Oracle Health Check Collections Manager Commands
  Use the `-cmupgrade` command to upgrade Oracle Health Check Collections Manager.
- Command-Line Options to Generate Password Protected Collection zip Files
  Use the list of commands to encrypt or decrypt diagnostic collection `zip` files.
- Applying Patch Between Releases
- Caching Discovery Data
  Use the list of commands to manage caching of discovery data.
- Configuring REST
  Use the list of commands to configure REST.
- Running Cluster Verification Utility (CVU) Compliance Checks
  Run Cluster Verification Utility (CVU) to perform system checks in preparation for installation, patch updates, or other system changes.
- Running Auto Start
  Use the list of commands to start or stop auto start.
- Application Continuity Command-Line Options
  Use the list of commands to configure Application Continuity.
- ZFS Storage Appliance Options
  Use the `-zfssa` command to run compliance checks on Oracle ZFS Storage Appliances.

A.1 Compliance Framework (Oracle ORAchk and Oracle EXAchk) Command-Line Options

Review the list of Compliance Framework (Oracle ORAchk and Oracle EXAchk) command-line options.

Syntax

```
$ orachk [options]
```

```
[-h] [-a] [-b] [-v] [-p] [-m] [-u] [-f] [-o]
[-clusternodes clusternames]
[-failedchecks previous_result]
[-nordbms]
[-output path]
[-dbnames dbnames]
[-localonly]
[-debug]
[-dbnone | -dball]
[-c]
[-upgrade | -noupgrade]
[-syslog]
[-skip_usr_def_checks]
[-checkfaileduploads]
[-uploadfailed all | comma-delimited list of collections]
[-fileattr [start | check | remove ] [-includedir path ] [-excludediscovery]
[-baseline path [-fileattronly]
[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-setdbupload all | db upload variable, for example,
RAT_UPLOAD_CONNECT_STRING, RAT_UPLOAD_PASSWORD]
[-unsetdbupload all | db upload variable, for example,
RAT_UPLOAD_CONNECT_STRING, RAT_UPLOAD_PASSWORD]
[-checkdbupload]
[-getdbupload]
[-cmupgrade]
[-sendemail "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-nopass]
[-noscore]
[-showpass]
[-show_critical]
[-diff Old Report New Report [-outfile Output HTML] [-force]]
[-merge report 1 report 2 [-force]]
[-tag tagname]
[-nodaemon]
[-profile asm | clusterware | corroborate | dba | ebs | emagent | emoms | em |
goldengate | hardware | maa | oam | oim | oud | ovn | peoplesoft | preinstall
| prepatch | security | siebel | solaris_cluster | storage | switch | sysadmin
| timesten | user_defined_checks |  zfs ]
[-excludeprofile asm | clusterware | corroborate | dba | ebs | emagent | emoms
| em | goldengate | hardware | maa | oam | oim | oud | ovn | peoplesoft |
preinstall | prepatch | security | siebel | solaris_cluster | storage | switch
| sysadmin | timesten | user_defined_checks | zfs ]
[-acchk -javahome path to jdk8
-asmhome path to asm-all-5.0.3.jar -appjar directory where jar files are
present for concrete class -apptrc directory where trace files are present
for coverage class]
[-check check ids | -excludecheck check ids]
[-zfsnodes nodes]
[-zfssa appliance names]
[-dbserial | -dbparallel [n] | -dbparallelmax]
[-idmpreinstall | -idmpostinstall | -idmruntime] [-topology topology.xml |
-credconfig credconfig] | -idmdbpreinstall | -idmdbpostinstall | -
idmdbruntime]
[-idm_config IDMCONFIG] [-idmdiscargs IDMDISCARGS]
[-idmhcargs IDMHCARGS | -h]
```

Syntax

```
$ exachk [options]


[-h] [-a] [-b] [-v] [-p] [-m] [-u] [-f] [-o]
[-clusternodes clusternames]
[-failedchecks previous_result]
[-nordbms]
[-output path]
[-dbnames dbnames]
[-localonly]
[-debug]
[-dbnone | -dball]
[-c]
[-upgrade | -noupgrade]
[-syslog] [-skip_usr_def_checks]
[-checkfaileduploads]
[-uploadfailed all | comma-delimited list of collections]
[-fileattr start | check | remove [-includedir path [-excludediscovery] [-
baseline path[-fileattronly]
[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-setdbupload all | db upload variable, for example,
RAT_UPLOAD_CONNECT_STRING, RAT_UPLOAD_PASSWORD]
[-unsetdbupload all | db upload variable, for example,
RAT_UPLOAD_CONNECT_STRING, RAT_UPLOAD_PASSWORD]
[-checkdbupload]
[-getdbupload]
[-cmupgrade] [-sendemail "NOTIFICATION_EMAIL=comma-delimited list of email
addresses"]
[-nopass]
[-noscore]
[-showpass]
[-show_critical]
[-diff Old Report New Report [-outfile Output HTML] [-force]]
[-merge report 1 report 2 [-force]]
[-tag tagname]
[-auto_restart -initsetup | -initdebugsetup | -initrmsetup | -initcheck | -h]
[-d start|start -debug|stop|status|info|stop_client|nextautorun|-h]
[-nodaemon]
```

```
[-unlockcells all | -cells comma-delimited list of names or IPs of cells] [-
lockcells all | -cells comma-delimited list of names or IPs of cells]
[-usecompute]
[-exadiff Exalogic collection1 Exalogic collection2]
[-vmguest ]
[-hybrid [-phy nodes]]
[-profile asm | bi_middleware | clusterware | compute_node | exatier1 |
control_VM | corroborate | dba | ebs | el_extensive | el_lite | el_rackcompare
| emagent | emoms | em | goldengate | hardware | maa | nimbula | obiee | ovn |
peoplesoft | platinum | preinstall | prepatch | security | siebel |
solaris_cluster | storage | switch | sysadmin | timesten | user_defined_checks
| virtual_infra]
[-excludeprofile asm | bi_middleware | clusterware | compute_node | exatier1 |
control_VM | corroborate | dba | ebs | el_extensive | el_lite | el_rackcompare
| emagent | emoms | em | goldengate | hardware | maa | nimbula | obiee | ovn |
peoplesoft | platinum | preinstall | prepatch | security | siebel |
solaris_cluster | storage | switch | sysadmin | timesten | user_defined_checks
| virtual_infra]
[-check check ids | -excludecheck check ids]
[-cells cells]
[-ibswitches switches]
[-torswitches]
[-extzfsnodes nodes]
[-dbserial | -dbparallel [n] | -dbparallelmax | -allserial]
[-allserial | -dbnodeserial |-cellserial | -switchserial]
```

A.2 Running Generic Compliance Framework (Oracle ORAchk and Oracle EXAchk) Commands

Review the list of generic Oracle ORAchk and Oracle EXAchk command options.

Syntax

```
[-a]
[-v]
[-debug]
[-nodaemon]
[-f]
[-upgrade]
[-noupgrade]
[-testemail all | "NOTIFICATION_EMAIL=comma-delimited list of email
addresses"]
[-sendemail “NOTIFICATION_EMAIL=comma-delimited list of email addresses"]
[-dbserial]
[-dbparallel [n]]
[-dbparallelmax]
```

Parameters

Table 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 Autonomous Health Framework compliance tools.
`-debug`	Runs in debug mode. The generated `.zip` file contains a debug log and other files useful for Oracle Support.
`-nodaemon`	Does not send commands to the daemon, usage is interactive.
`-f`	Runs Offline. The tools perform health checks on the data already collected from the system.
`-upgrade`	Forces an upgrade of the version of the tools being run.
`-noupgrade`	`-noupgrade` is for when you have the latest version in `RAT_UPGRADE_LOC` and do not yet want to upgrade. Adding `-noupgrade` without having the latest version in `RAT_UPGRADE_LOC` will still prompt you to download the latest version.
`-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.
`-dbparallelmax`	Runs the `SQL`, `SQL_COLLECT`, and `OS` health checks in parallel, using the maximum number of child processes.

A.3 Controlling the Scope of Checks

Use the list of commands to control the scope of checks.

Syntax

```
[-b]
[-p]
[-m]
[-u -o pre]
[-u -o post]
[-clusternodes nodes]
[-failedchecks previous_result]
```

```
[-nordbms]
[-dbnames db_names]
[-dbnone]
[-dball]
[-localonly]
[-cells cells]
[-ibswitches switches]
[-profile profile]
[-excludeprofile profile]
[-check check_id]
[-excludecheck check_id]
[-skip_usr_def_checks]
```

Parameters

Table 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 Oracle Database.
-u -o post	Runs the post-upgrade checks for Oracle Clusterware and Oracle Database.
-clusternodes *nodes*	Specify a comma-delimited list of node names to run only on a subset of nodes.
-failedchecks *previous_result*	Runs only checks from the *presious_result*, which had failed.
-nordbms	Runs Oracle Grid Infrastructure checks only in environments with no Oracle Database checks performed.
-dbnames *db_names*	Specify a comma-delimited list of database names to run only on a subset of databases.
-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.

Table A-2 (Cont.) Scope of Checks

Command	Description
`-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.4 Managing the Report Output

Use the list of commands to manage compliance checks 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.
`-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.5 Uploading Results to Database

Use the list of commands 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 compliance 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 compliance 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.
`-checkdbupload`	Checks if the variables are set correctly for uploading the compliance check run results to the database.
`-getdbupload`	Prints the variables with their values from wallet for uploading the compliance 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.6 Controlling the Behavior of the Daemon

Use the list of commands 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 start -ords]
[-d start -ords ords_path]
[-d start -ords [-ordscollectionretention size_mbs]]
[-d info]
```

```
[-id id] -d nextautorun
[-initsetup]
[-initrmsetup]
[-initcheck]
```

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 start -ords`	Starts the daemon to serve Oracle Rest Data Service (ORDS) API requests. Run the `-ordssetup` command before starting the Oracle ORAchk daemon to run the Oracle Rest Data Service (ORDS).
`-d start -ords ords_path`	Starts the daemon to serve Oracle Rest Data Service (ORDS) API requests. Requires Oracle Rest Data Service (ORDS) to be up and running at the specified path, `ords_path`.
`-d start -ords [-ordscollectionretention` *size_mbs*`]`	Starts the daemon to serve Oracle Rest Data Service (ORDS) API requests, and sets the ORDS collection retention to the specified size, *size_mbs* MB. The default collection retention value is 1024 MB.
`-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.

A.7 Tracking File Attribute Differences

Use the list of commands to track file attribute differences.

Parameters

Table A-6 File Attribute Differences

Option	Description
`-fileattr start`	Takes file attributes snapshot of discovered directories and stores the snapshot in the output directory. By default, the tool takes snapshot of Oracle Grid Infrastructure home and all the installed database homes. If the user doesn't own a particular directory, then the tool does not take snapshot of the directory.
`-fileattr check`	Takes a recent snapshot of discovered directories and compares with the previous snapshot
`-fileattr remove`	Removes the file attribute snapshots and related files.

`-fileattr [start|check] -includedir` *`directories`*	Includes the directories specified at the command-line 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. `orachk -fileattr start -includedir "/root/home,/etc" -excludediscovery`
`-fileattr check -baseline` *`baseline snapshot path`*	For example: `orachk -fileattr check -baseline "/tmp/Snapshot"`
`-fileattr -check -fileattronly`	Performs file attributes check and exits Oracle ORAchk. `orachk -fileattr check -fileattronly`

A.8 Running Oracle Health Check Collections Manager Commands

Use the `-cmupgrade` command to upgrade Oracle Health Check Collections Manager.

Table A-7 Oracle Health Check Collections Manager Commands

Command	Description

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

A.9 Command-Line Options to Generate Password Protected Collection zip Files

Use the list of commands to encrypt or decrypt diagnostic collection `zip` files.

Table A-8 Encrypt and Decrypt Diagnostic Collection zip Files

Option	Description

`orachk -d start -encryptzip`
`exachk -d start -encryptzip`
	Starts the daemon with `-encryptzip` option.
	The daemon prompts for a password when it starts. The daemon then encrypts the subsequent on-demand and scheduled runs collections with that password.
	Note: When `-encryptzip` is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection `zip` file deletes the collections directory.

Table A-8 (Cont.) Encrypt and Decrypt Diagnostic Collection zip Files

Option	Description
orachk [-option *value*] -encryptzip exachk [-option *value*] -encryptzip	Encrypts the run result. Prompts for the password, and encrypts the collections created at the end of the run with that password. You can use -encryptzip with other Oracle ORAchk and Oracle EXAchk options that generate a collection. For example: orachk -profile *profile-name* -encryptzip orachk -profile sysadmin -encryptzip orachk -check *check-id* -encryptzip orachk -check D47661C55B1A291AE0431EC0E50A5C53 -encryptzip **Note:** When -encryptzip is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection zip file deletes the collections directory.
orachk -encryptzip *zip_file* exachk -encryptzip *zip_file*	Encrypts the already generated collection. Prompts for the password, encrypts the zip file specified with that password, and then renames the collections as, for example, orachk_*host*_*db*_encrypted_*date*_*time*.zip. **Note:** When -encryptzip is passed, Oracle ORAchk and Oracle EXAchk after successfully encrypting the diagnostic collection zip file deletes the collections directory.
orachk -decryptzip *zip_file* exachk -decryptzip *zip_file*	Decrypts the encrypted collection. Prompts for the password, decrypts the zip file specified with that password, and then renames the collections as, for example, orachk_*host*_*db*_*date*_*time*.zip.

A.10 Applying Patch Between Releases

Use the list of commands to manage patches.

Syntax

```
orachk -applypatch orachk_bug_id.zip
exachk -applypatch exachk_bug_id.zip


orachk -querypatch all
exachk -querypatch all
orachk -querypatch bug_id
exachk -querypatch bug_id


orachk -rollbackpatch bug_id
exachk -rollbackpatch bug_id
```

Table A-9 Managing Patches

Command	Description
`-applypatch`	Applies a new patch for the specified bug ID.
`-querypatch`	Lists the details of all of the installed patches or for the specified bug ID.
`-rollbackpatch`	Rolls back the applied patch to its previous state, the state at which the patch was applied.

A.11 Caching Discovery Data

Use the list of commands to manage caching of discovery data.

Syntax

```
orachk -discovery -discoverydir location
exachk -discovery -discoverydir location


orachk -checkdiscovery
exachk -checkdiscovery


orachk -usediscovery -discoverydir location
exachk -usediscovery -discoverydir location


orachk -rediscovery
exachk -rediscovery


orachk -rmdiscovery
exachk -rmdiscovery
```

Table A-10 Manage Caching of Discovery Data

Command	Description
`-discovery`	Caches discovery data, which Oracle ORAchk and Oracle EXAchk can use for future runs. `-discoverydir`: Specify the location to store the discovery data.
`-checkdiscovery`	Verifies the discovery data.
`-usediscovery`	Uses the discovery data. `-discoverydir`: Specify the location where you have cached the discovery data.
`-rediscovery`	Refreshes the cache discovery data.
`-rmdiscovery`	Removes the cached discovery data.

A.12 Configuring REST

Use the list of commands to configure REST.

Syntax

```
orachk -ordssetup [dir [-configdir dir_to_store configuring ORDS]] [-
ordshomedir any_directory_with_write_permission]
```

```
exachk -ordssetup [dir [-configdir dir_to_store configuring ORDS]] [-
ordshomedir any_directory_with_write_permission]
```

```
orachk -ordsrmsetup
exachk -ordsrmsetup
```

```
orachk -ordscheck
exachk -ordscheck
```

Table A-11 Configure REST

Command	Description
`-ordssetup`	Sets up ORDS on the target host. `dir`: The directory that contains the `ords.war` file. `-configdir`: Optional directory that you can specify to store the ORDS configuration files. If you do not specify the optional directory, then the configuration files are stored in the directory that contains the `ords.war` file. `-ordshomedir`: Optional directory that you can specify if `root` does not have the privilege to run the `useradd` command to create the default home directory. When you specify `-ordshomedir`, user home will be the path passed along with `-ordshomedir`.
`-ordsrmsetup`	Removes ORDS setup. Running the command stops the daemon if running and deletes the ORDS user's home directory if no collections are found. If collections from previous runs are found, then the command prompts the user before a decision is made to remove the setup or not.
`-ordscheck`	Running the command lets the user know if ORDS is setup or not. If ORDS is setup, then the command prints the URL to use to submit runs using REST APIs.

A.13 Running Cluster Verification Utility (CVU) Compliance Checks

Run Cluster Verification Utility (CVU) to perform system checks in preparation for installation, patch updates, or other system changes.

> **Note:**
>
> You can run CVU check as `root` or a non-root user. Currently, running CVU checks are limited to Linux and Solaris.

CVU is integrated into Oracle ORAchk and Oracle EXAchk. By default,

- CVU health checks are run when you run Oracle ORAchk on Oracle RAC, Oracle Restart, and Oracle Database Appliance (ODA).
- A full Oracle EXAchk run includes CVU health checks.

Oracle ORAchk and Oracle EXAchk include the Cluster Verification Utility (CVU) compliance check results in the following reports:

- Oracle RAC Assessment Report
- Oracle RAC Upgrade Readiness Report
- Oracle Exadata Assessment Report

When you run the `-profile preinstall` command, preinstallation related CVU checks are run for Oracle Database and Oracle Clusterware.

When you run Oracle ORAchk and Oracle EXAchk in pre-upgrade mode, pre-upgrade related CVU checks are run for Oracle Database and Oracle Clusterware.

When you run Oracle ORAchk and Oracle EXAchk in normal or pre-upgrade mode, CVU will only be used if the tools find CVU is available, recent and valid for the situation you are using it in.

These are the checks performed to validate CVU:

- CVU exists in `ahf_dir/common/cvu` directory or the path specified using the `-cvuhome` option.
- The CVU pack is less than 180 days. Note that you can modify this value by setting the `RAT_STALE_DAYS=`*`n`* environment variable.
- If the CVU version is equal or higher than the CRS version.
- If the CVU version is equal or higher than the upgrade target version.

If you are running as `root` and one of the above validations fail, then the tools will prompt to download the latest CVU from My Oracle Support. If My Oracle Support credentials are already configured in the wallet, then these will be used. If not, then the tools will prompt for My Oracle Support username and password.

After downloading a new CVU pack the tools automatically distribute this to all nodes in the cluster. By default this cluster distribution is done through the TFA secure socket connection. Distribution through the TFA secure socket connection is only possible if:

- The tools were installed through a full installation and not using the `-extract` option, or installed as non-root.
- The Oracle Trace File Analyzer daemon has not been shutdown.

CVU pack cluster distribution can be done through passwordless SSH if the originating Oracle ORAchk or Oracle EXAchk command was run with the `-usessh` option for example:

```
orachk -usessh
orachk -preupgrade -usessh


exachk -usessh
exachk -preupgrade -usessh
```

You can prevent the prompting for CVU upgrade using any one of the following options:

- Set the `RAT_NOCVU_UPGRADE` environment variable to 1, for example, `RAT_NOCVU_UPGRADE=1`.
- Set the `RAT_NOUPGRADE` environment variable to 1, for example, `RAT_NOUPGRADE=1`.
- Run Oracle ORAchk and Oracle EXAchk with the `-noupgrade` option.

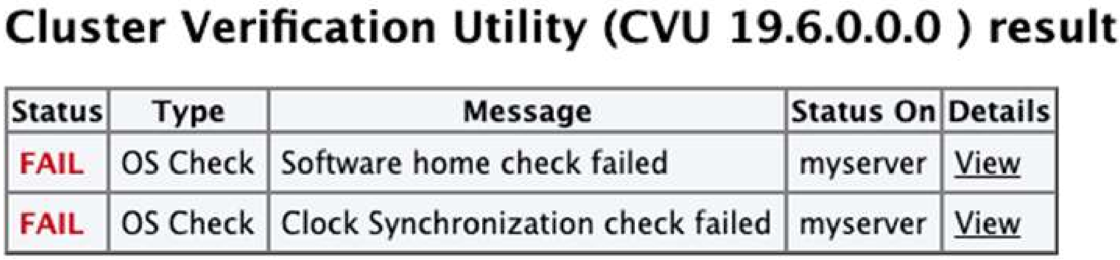
For example:

```
orachk -noupgrade
orachk -preupgrade -noupgrade
```

```
exachk -noupgrade
exachk -preupgrade -noupgrade
```

Oracle ORAchk and Oracle EXAchk report includes the CVU version and the CVU checks result.

Figure A-1   CVU Result

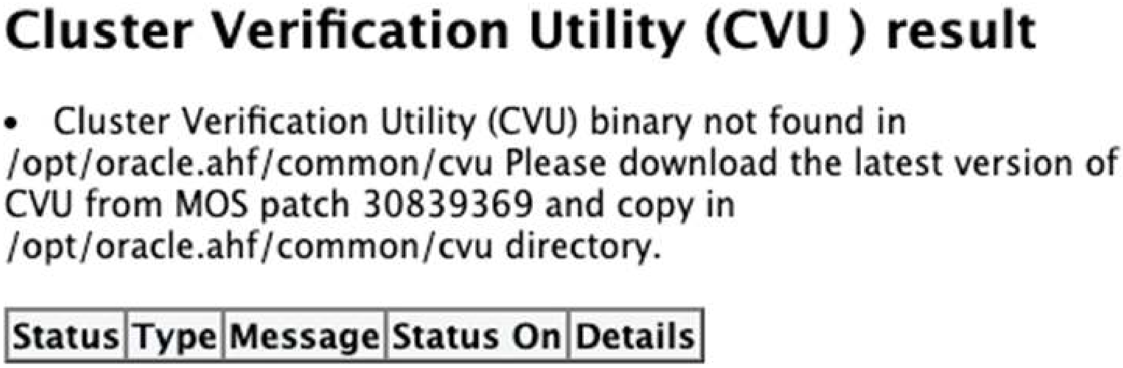


Cluster Verification Utility (CVU 19.6.0.0.0) result

Status	Type	Message	Status On	Details
FAIL	OS Check	Software home check failed	myserver	View
FAIL	OS Check	Clock Synchronization check failed	myserver	View

If CVU pack is not found or if the latest version is not available, then Oracle ORAchk and Oracle EXAchk logs the message and add an entry within the report.

Figure A-2   No CVU Result



Cluster Verification Utility (CVU) result

- Cluster Verification Utility (CVU) binary not found in /opt/oracle.ahf/common/cvu Please download the latest version of CVU from MOS patch 30839369 and copy in /opt/oracle.ahf/common/cvu directory.

Status	Type	Message	Status On	Details

Syntax

```
orachk [-cvuhome] [-cvuonly] [-includecvu] [-excludecvu]
exachk [-cvuhome] [-cvuonly] [-includecvu] [-excludecvu]
```

Parameters

Table A-12 Running CVU Compliance Checks

Option	Description

`-cvuhome`	Specify the location of the zipped file `cvupack.zip` or the directory where you have unzipped the `cvupack.zip` file.
	`orachk -cvuhome` *gi_home*
	`exachk -cvuhome` *gi_home*
	`orachk -cvuhome` *path to cvu_zip*
	`exachk -cvuhome` *path to cvu_zip*
	`orachk -cvuhome` *path to unzipped cvupack*
	`exachk -cvuhome` *path to unzipped cvupack*
	`orachk -cvuhome` *location_of_file_or_directory*
	`exachk -cvuhome` *location_of_file_or_directory*
	`orachk -profile preinstall -cvuhome` *location_of_file_or_directory*
	`exachk -profile preinstall -cvuhome` *location_of_file_or_directory*
	`orachk -preupgrade -cvuhome` *location_of_file_or_directory*
	`exachk -preupgrade -cvuhome` *location_of_file_or_directory*
	For example:
	`orachk -cvuhome /tmp/cvupack.zip`
	`orachk -cvuhome /tmp/cvupack`
	`orachk -profile preinstall -cvuhome /tmp/ cvupack.zip`
	`orachk -profile preinstall -cvuhome /tmp/cvupack`
	`orachk -preupgrade -cvuhome /tmp/cvupack.zip`
	`orachk -preupgrade -cvuhome /tmp/cvupack`

`-cvuonly`	Use the `-cvuonly` command option to run only the CVU checks. Running the `-cvuonly` command does not run Oracle ORAchk and Oracle EXAchk related compliance checks.
`-excludecvu`	Use the `-excludecvu` command option to exclude CVU checks.

Reviewing Cluster Verification Utility (CVU) Output

By default, a full Oracle EXAchk run calls CVU and displays the results in a separate section of the report. To review the CVU output, run Oracle EXAchk and review the provided report. Also by default, only the FAIL items are displayed, so the expected output (all PASS results) in the Oracle EXAchk report displays only the header information similar to:

```
Cluster Verification Utility (CVU 19.4.0.0.0 ) result
Status    Type    Message    Status On    Details
```

If you wish to view the specific CVU verifications, select **PASS** or **ALL** in the Oracle EXAchk report header section, and you will see output similar to:

```
Cluster Verification Utility (CVU 19.4.0.0.0 ) result
Status    Type    Message    Status On    Details
PASS   OS Check   Node Connectivity check passed   random01client01   View
PASS   OS Check   Multicast or broadcast check check passed
random01client01   View
PASS   OS Check   Time zone consistency check passed   random01client01   View
PASS   OS Check   Cluster Manager Integrity check passed   random01client01
View
PASS   OS Check   Cluster Integrity check passed   random01client01   View
PASS   OS Check   CRS Integrity check passed   random01client01   View
PASS   OS Check   Node Application Existence check passed
random01client01   View
PASS   OS Check   Single Client Access Name (SCAN) check passed
random01client01   View
PASS   OS Check   OLR Integrity check passed   random01client01   View
PASS   OS Check   ASM Integrity check passed   random01client01   View
PASS   OS Check   User Not In Group "root": grid check passed
random01client01   View
PASS   OS Check   Clock Synchronization check passed   random01client01   View
PASS   OS Check   VIP Subnet configuration check check passed
random01client01   View
PASS   OS Check   Network configuration consistency checks check passed
random01client01   View
PASS   OS Check   Package: psmisc-22.6-19 check passed   random01client01
View
PASS   OS Check   File system mount options for path GI_HOME check passed
random01client01   View
PASS   OS Check   ACFS device special file check passed   random01client01
View
```

In this section of the report, click the **View** link to view more details. For example, in the **Node Connectivity check passed** entry above:

```
Description   This is a prerequisite condition to test whether connectivity
exists amongst all the nodes.
The connectivity is being tested for the subnets
"98.450.312.0,98.450.312.0,98.450.312.0,99.475.0.0"

Links   None

Needs attention on   -
```

```
Passed on   random01client01

Status on random01client01:
PASS => Node Connectivity check passed
```

If there are any CVU issues reported, then the default report will show an expanded table similar to the following:

```
Cluster Verification Utility (CVU 19.4.0.0.0 ) result
Status    Type    Message    Status On    Details
FAIL   OS Check   Node Connectivity check failed   random01client01   View
```

Examine the additional information in the **View** detail section for root cause and take appropriate corrective action.

> **Note:**
>
> For additional information on the Cluster Verification Utility, see *Cluster Verification Utility Referece* section of the appropriate *Clusterware Administration and Deployment Guide* for the installed Oracle Database version.

> **Note:**
>
> If you wish to review the CVU output without a full Oracle EXAchk run after completing the corrective actions, then as `root` run the following command in the directory in which Oracle EXAchk was installed:
>
> ```
> exachk -cvuonly
> ```

Related Topics

- Cluster Verification Utility Reference

A.14 Running Auto Start

Use the list of commands to start or stop auto start.

Table A-13   Auto start

Option	Description
`-autostart`	Configures auto start. You must run this command as `root`. The daemon restarts at 1 am every day to discover any environment changes. The daemon runs a full local Oracle ORAchk check once every week at 3 am, and a partial run of the most impactful checks at 2 am every day through the `oratier1` or `exatier1` profiles. The daemon automatically purges the `oratier1` or `exatier1` profile run that runs daily, after a week. The daemon also automatically purges the full local run after 2 weeks. You can change the daemon settings after enabling auto start. **Note:** Daemon mode is supported only on the Linux and Solaris operating systems.
`-autostop`	Removes auto start configuration. You must run this command as `root`.

A.15 Application Continuity Command-Line Options

Use the list of commands to configure Application Continuity.

Table A-14   Application Continuity Command-Line Options

Command-Line Argument	Shell Environment Variable	Usage
`-asmhome jarfilename`	`RAT_AC_ASMJAR`	This must point to a version of `asm-all-5.0.3.jar` that you download from ASM Home Page.
`-javahome JDK8dirname`	`RAT_JAVA_HOME`	This must point to the `JAVA_HOME` directory for a JDK8 installation.
`-appjar dirname`	`RAT_AC_JARDIR`	To analyze the application code for references to Oracle concrete classes, this must point to the parent directory name for the code. The program analyzes `.class` files, and recursively `.jar` files and directories. To analyze the coverage, specify a directory name that contains one or more database server trace files. The trace directory is generally, `$ORACLE_BASE/diag/rdbms/$ORACLE_UNQNAME/$ORACLE_SID/trace`
NONE	`RAT_ACTRACEFILE_WINDOW`	When scanning the trace directory, this optional value limits the analysis to scanning to files created in the most recent specified number of days

Example A-1   Application Continuity Command-Line Options

```
$ orachk -asmhome /tmp/asm-all-5.0.3.jar -javahome /tmp/jdk1.8.0_40 -
apptrc $ORACLE_BASE/diag/rdbms/$ORACLE_SID/trace 3
```

Related Topics

- ASM - Home Page

A.16 ZFS Storage Appliance Options

Use the `-zfssa` command to run compliance checks on Oracle ZFS Storage Appliances.

Table A-15   ZFS Storage Appliance Options

Option	Description
`-zfssa` *`node`*	Runs Oracle ORAchk only on selected ZFS appliance nodes, where *node* is a comma-delimited list of ZFS Storage Appliance names. For example: `orachk -zfssa node1,node2`

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.

- oclumon debug
  Use the `oclumon debug` command to set the log level for the Cluster Health Monitor services.
- oclumon dumpnodeview
  Use the `oclumon dumpnodeview` command to view log information from the system monitor service in the form of a node view.
- oclumon dumpnodeview local
  Use the `oclumon dumpnodeview local` command to view log information from the system monitor service in the form of a node view.
- oclumon manage
  Use the `oclumon manage` command to view and change configuration information from the system monitor service.
- oclumon version
  Use the `oclumon version` command to obtain the version of Cluster Health Monitor that you are using.

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

Cluster Health Monitor (OS), Release 20.0.0.0.0
Version             : 20.3.0.0.0
NODEVIEW Version    : 19.03
Label Date          : 200116
```

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

> **Note:**
>
> The `oclumon dumpnodeview` commands work only if GIMR or MGMTDB is configured and is in ONLINE state.

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 [-v2]]
[-process]
[-procag]
[-device]
[-filesystem]
[-nic]
[-advm]
[-protocols]
[-cpu]
[-topconsumer]
[-asminst_db]
[-nfs]]
[-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.

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

Parameter	Description
`-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.
`-i `*`interval`*	Specify a collection interval, in five-second increments.
`-v`	Displays verbose node view output.
`-system[-v2], -topconsumer, -process, -cpu, -procag, -device, -filesystem, -nic, -protocols, -advm, -asminst_db, -nfs`	Dumps each specified node view parts.
`-v2`	Dumps nodeview part output with version2 outlook. Currently available with the "system" part.
`-format "`*`format type`*`"`	Specify the output format. "*format type*" can be `legacy`, `tabular`, `json`, 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.

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

Parameter	Description
`-procag`	Outputs the process of the node view, aggregated by category: • DBBG (DB backgrounds) • DBFG (DB foregrounds) • CLUST (Cluster) • OTHER (other processes) **Note:** `-procag` is currently available only on Linux, Solaris, and AIX. It is not supported on Microsoft Windows systems.
`-h`	Displays online help for the `oclumon dumpnodeview` command.

Usage Notes

- In certain circumstances, data can be delayed for some time before the command replays the data.

  For example, the `crsctl stop cluster -all` command can cause data delay. After running `crsctl start cluster -all`, it may take several minutes before `oclumon dumpnodeview` shows any data collected during the interval.
- The default is to continuously dump node views. To stop continuous display, use Ctrl+C on Linux and Microsoft Windows.
- Both the local system monitor service (`osysmond`) and the cluster logger service (`ologgerd`) must be running to obtain node view dumps.
- The `oclumon dumpnodeview` command displays only 127 CPUs of the CPU core, omitting a CPU at random from the list.

Metric Descriptions

This section includes descriptions of the metrics in each of the seven views that comprise a node view listed in the following tables.

Table B-3 oclumon dumpnodeview SYSTEM View Metric Descriptions

Metric	Description
`#pcpus`	Number of physical CPUs.
`#cores`	Number of CPU cores in the system.
`#vcpus`	Number of logical compute units.
`cpuht`	CPU hyperthreading enabled (Y) or disabled (N).
`chipname`	Name of the CPU vendor.
`cpu`	Average CPU utilization per processing unit within the current sample interval (%).
`cpuusage`	Total CPU usage = `cpusystem` + `cpuuser` + `cpunice` Percentage of over all CPU cores. 100% indicates that all cores are spent for that metric.
`cpusystem`	CPU used by processes in kernel mode.
`cpuuser`	CPU used by normal processes in user mode.
`cpunice`	CPU used by "niced" processes (low priority).

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

Metric	Description
`cpuiowait`	CPU waiting for I/O.
`cpusteal`	Virtual CPU waiting for physical CPU to be freed by other VM.
`cpuq`	Number of processes waiting in the run queue within the current sample interval.
`physmemfree`	Amount of free RAM (KB).
`physmemtotal`	Amount of total usable RAM (KB).
`shmem`	Shared memory.
`mcache`	Amount of physical RAM used for file buffers plus the amount of physical RAM used as cache memory (KB). On Microsoft Windows systems, this is the number of bytes currently being used by the file system cache. **Note:** This metric is not available on Oracle Solaris.
`swapfree`	Amount of swap memory free (KB)
`swaptotal`	Total amount of physical swap memory (KB)
`hugepagetotal`	Total size of huge in KB **Note:** This metric is not available on Solaris or Microsoft Windows systems.
`hugepagefree`	Free size of huge page in KB **Note:** This metric is not available on Solaris or Microsoft Windows systems.
`hugepagesize`	Smallest unit size of huge page **Note:** This metric is not available on Solaris or Microsoft Windows systems.
`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.

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

Metric	Description
`#procs_blocked`	Number of processes currently blocked waiting for I/O.
`rtprocs`	Number of real-time processes.
`rtprocsoncpu`	The current number of real-time processes running on the CPU.
`#fds`	Number of open file descriptors. *or* Number of open handles on Microsoft Windows.
`#sysfdlimit`	System limit on the number of file descriptors. **Note:** This metric is not available on either Solaris or Microsoft Windows systems.
`#disks`	Number of disks.
`#nics`	Number of network interface cards.
`nicErrors`	Average total network error rate within the current sample interval (errors per second).
`#nfs`	Number of network file system.
`loadavg1 loadavg5 loadavg15`	Load average (average number of jobs in the run queue or waiting for disk I/O) of the last 1, 5, 15 minutes.

Table 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.
`ppid`	PID of the parent process. For example, if process 1 spawns process 2, then ppid of process 2 is pid of process 1.
`cumulative_cpu`	The total amount of CPU time this process is scheduled to run since it started. The total amount of CPU time spent for this process so far is measured in micro seconds.
`#procfdlimit`	Limit on number of file descriptors for this process. **Note:** This metric is not available on Microsoft Windows, AIX, and HP-UX systems.
`cpuusage`	Process CPU utilization (%). **Note:** The utilization value can be up to 100 times the number of processing units.
`vmem`	Process virtual memory usage (KB).
`privmem`	Process private memory usage (KB).
`shmem, shm, and sharedmem`	Process shared memory usage (KB). **Note:** This metric is not available on Microsoft Windows, Solaris, and AIX systems. It is supported only on Linux systems.
`workingset`	Working set of a program (KB) **Note:** This metric is only available on Microsoft Windows.

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

Metric	Description
`#fd`	Number of file descriptors open by this process. *or* Number of open handles by this process on Microsoft Windows.
`#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).
`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).

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

Metric	Description
`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.
`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.

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

Metric	Description
TCPRetraSeg	Total number of TCP segments retransmitted.
UDPUnkPort	Total number of received UDP datagrams for which there was no application at the destination port.
UDPRcvErr	Number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

Table 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[%]","cpuus
er[%]",
"cpunice[%]","cpuiowait[%]","cpusteal[%]","cpuq","physmemfree[KB]","physmemtot
```

```
al[KB]",
"mcache[KB]","swapfree[KB]","swaptotal[KB]","hugepagetotal","hugepagefree","hu
gepagesize",
"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","#nicE
rrors"
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 dumpnodeview local

Use the `oclumon dumpnodeview local` command to view log information from the system monitor service in the form of a node view.

Usage Notes

> **Note:**
>
> The `oclumon dumpnodeview local` command has no dependency on GIMR or MGMTDB and can return Cluster Health Monitor data irrespective of GIMR being configured.

Syntax

```
dumpnodeview local [[([(-system | -protocols | -v)] |
     [(-cpu | -process | -procagg | -device | -nic | -filesystem | -nfs)
     [-detail] [-all] [-sort <metric_name>] [-filter <string>] [-head
<rows_count>] [-i <seconds>]])
     [([-s <start_time>  -e <end_time>] | -last <duration>)]] |
     [-inputDataDir <absolute_path> -logDir <absolute_path>]
     [-h]]
```

Parameters

Table B-10 oclumon dumpnodeview local Command Parameters

Parameter	Description
`local`	Cluster Health Monitor local dump. Dumps metrics from node local Cluster Health Monitor repository.
`-system`	Dumps system metrics. For example: `oclumon dumpnodeview local -system` .
`-cpu`	Dumps CPU metrics. For example: `oclumon dumpnodeview local -cpu` .
`-process`	Dumps process metrics. For example: `oclumon dumpnodeview local -process` .

Table B-10 (Cont.) oclumon dumpnodeview local Command Parameters

Parameter	Description
`-procagg`	Dumps process aggregate metrics. For example: `oclumon dumpnodeview local -procagg` .
`-device`	Dumps disk metrics. For example: `oclumon dumpnodeview local -device` .
`-nic`	Dumps network interface metrics. For example: `oclumon dumpnodeview local -nic` .
`-filesystem`	Dumps filesystem metrics. For example: `oclumon dumpnodeview local -filesystem` .
`-nfs`	Dumps NFS metrics. For example: `oclumon dumpnodeview local -nfs` .
`-protocols`	Dumps network protocol metrics, cumulative values from system start. For example: `oclumon dumpnodeview local -protocols` .
`-v`	Displays verbose node view output. For example: `oclumon dumpnodeview local -v` .
`-h, --help`	Displays the command-line help and exits.

Table B-11 oclumon dumpnodeview local Command Flags

Flag	Description
`-detail`	Use this option to dump detailed metrics. Applicable to the `-process` and `-nic` options. For example: `oclumon dumpnodeview local -process -detail` .
`-all`	Use this option to dump the node views of all entries. Applicable to the `-process` option. For example: `oclumon dumpnodeview local -process -all` .
`-head` *`rows_count`*	Use this option to dump the node view of the specified number of metrics rows in the result. Applicable to the `-process` option. Default is set to 5. For example: `oclumon dumpnodeview local -process -head 7` .
`-sort` *`metric_name`*	Use this option to sort based on the specified metric name, supported with the `-process`, `-device`, `-nic`, `-cpu`, `-procagg`, `-filesystem`, `-nfs` options. For example: `oclumon dumpnodeview local -device -sort "ioR"` .
`-i` *`seconds`*	Display data separated by the specified interval in seconds. Must be a multiple of 5. Applicable to continuous mode query. For example: `oclumon dumpnodeview local -device -i 5`

Table B-11 (Cont.) oclumon dumpnodeview local Command Flags

Flag	Description
`-filter` *`string`*	Use this option to search for a filter string in the Name column of the respective metric. For example, `-process -filter "ora"` will display the process metrics, which contain "`ora`" substring in their name. Supported with the `-process`, `-device`, `-nic`, `-cpu`, `-procagg`, `-filesystem`, `-nfs` options. For example: `oclumon dumpnodeview local -process -filter "ora"` .
`-show_all_sample_with_filter`	All samples where filter doesn't matches will also show in the output . Can be used only with -filter option. For example: `oclumon dumpnodeview local -filter` *`filter_criteria`* `-show_all_sample_with_filter`

Table B-12 oclumon dumpnodeview local Command Log File Directories

Directory	Description
`-inputDataDir` *`absolute_dir_path`*	Specifies absolute path of the directory that contains JSON logs files. For example: `oclumon dumpnodeview local -cpu -inputDataDir` *`absolute_path`*
`-logDir` *`absolute_log_dir_path`*	Specifies absolute path of the directory, which will contain the script run logs. For example: `oclumon dumpnodeview local -cpu -inputDataDir` *`absolute_path`* `-logDir` *`absolute_log_dir_path`*

Table B-13 oclumon dumpnodeview local Command Historical Query Options

Flag	Description

`-s` *`start_time`*
`-e` *`end_time`*
	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 the `YYYY-MM-DD HH24:MM:SS` format surrounded by double quotation marks (`""`).
	Specify these two options together to obtain a range.
	For example:
	`oclumon dumpnodeview local -cpu -s "2019-07-10 03:40:25" -e "2019-07-10 03:45:25"`

`-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. Specifying "*`00:45:00`*" will dump metrics for the last 45 minutes. For example: `oclumon dumpnodeview local -nic -last "00:45:00"` .

B.4 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-14 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.5 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), Release 20.0.0.0.0
Version : 20.3.0.0.0
```

C
Managing the Cluster Resource Activity Log

Oracle Clusterware stores logs about resource failures in the cluster resource activity log, which is located in the Grid Infrastructure Management Repository.

Failures can occur as a result of a problem with a resource, a hosting node, or the network.

The cluster resource activity log provides precise and specific information about a resource failure, separate from diagnostic logs. The cluster resource activity log also provides a unified view of the cause of resource failure.

Use the following commands to manage and view the contents of the cluster resource activity log:

- crsctl query calog
  Query the cluster resource activity logs matching specific criteria.
- crsctl get calog maxsize
  To store Oracle Clusterware-managed resource activity information, query the maximum space allotted to the cluster resource activity log.
- crsctl get calog retentiontime
  Query the retention time of the cluster resource activity log.
- crsctl set calog maxsize
  Configure the maximum amount of space allotted to store Oracle Clusterware-managed resource activity information.
- crsctl set calog retentiontime
  Configure the retention time of the cluster resource activity log.

C.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 C-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 C-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.

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

Field	Description	Use Case
`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.
`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.

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

Field	Description	Use Case
`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.
`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 :
```

C.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.
```

C.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.
```

C.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
```

C.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
```

D

chactl Command Reference

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

- chactl monitor
  Use the `chactl monitor` command to start monitoring all the instances of a specific Oracle Real Application Clusters (Oracle RAC) database using the current set model.
- chactl unmonitor
  Use the `chactl unmonitor` command to stop monitoring all the instances of a specific database.
- chactl status
  Use the `chactl status` command to check monitoring status of the running targets.
- chactl config
  Use the `chactl config` command to list all the targets being monitored, along with the current model of each target.
- chactl calibrate
  Use the `chactl calibrate` command to create a new model that has greater sensitivity and accuracy.
- chactl query diagnosis
  Use the `chactl query diagnosis` command to return problems and diagnosis, and suggested corrective actions associated with the problem for specific cluster nodes or Oracle Real Application Clusters (Oracle RAC) databases.
- chactl query model
  Use the `chactl query model` command to list all Oracle Cluster Health Advisor models or to view detailed information about a specific Oracle Cluster Health Advisor model.
- chactl query repository
  Use the `chactl query repository` command to view the maximum retention time, number of targets, and the size of the Oracle Cluster Health Advisor repository.
- chactl query calibration
  Use the `chactl query calibration` command to view detailed information about the calibration data of a specific target.
- chactl remove model
  Use the `chactl remove model` command to delete an Oracle Cluster Health Advisor model along with the calibration data and metadata of the model from the Oracle Cluster Health Advisor repository.
- chactl rename model
  Use the `chactl rename model` command to rename an Oracle Cluster Health Advisor model in the Oracle Cluster Health Advisor repository.
- chactl export model
  Use the `chactl export model` command to export Oracle Cluster Health Advisor models.
- chactl import model
  Use the `chactl import model` command to import Oracle Cluster Health Advisor models.

- chactl set maxretention
  Use the `chactl set maxretention` command to set the maximum retention time for the diagnostic data.
- chactl resize repository
  Use the `chactl resize repository` command to resize the tablespace of the Oracle Cluster Health Advisor repository based on the current retention time and the number of targets.

D.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 D-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.

D.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
```

D.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.

D.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
```

D.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.

D.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.

D.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
  ...
  ```

D.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
```

D.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%
```

D.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.

D.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.

D.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
```

D.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 D-1   Example

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

D.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.

D.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
```

E
Oracle Autonomous Health Framework Command-Line and Shell Options

TFACTL is the command-line interface for Oracle Trace File Analyzer.

TFACTL provides a command-line and shell interface to Oracle Autonomous Health Framework commands for:

- Administration
- Summary and analysis
- Diagnostic collection

The `tfactl` commands that you can run depends on your access level.

- You need `root` access or `sudo` access to `tfactl` to run administration commands.
- Run a subset of commands as:
  - An Oracle Database home owner or Oracle Grid Infrastructure home owner
  - A member of `OS DBA` or `ASM` groups

  You gain access to summary, analysis, and diagnostic collection functionality by running the commands as an Oracle Database home owner or Oracle Grid Infrastructure home owner.

To grant other users access to `tfactl`:

```
tfactl access
```

To use `tfactl` as a command-line tool:

```
tfactl command [options]
```

To use `tfactl` as a shell interface:

```
tfactl
```

Once the shell starts enter commands as needed.

```
$ tfactl

tfactl>
```

Append the `-help` option to any of the `tfactl` commands to obtain command-specific help.

```
$ tfactl command -help
```

- Oracle Autonomous Health Framework Installation Command-Line Options
  Understand the options that you can supply to the Oracle Autonomous Health Framework installer script to customize the installation.
- Running Oracle Trace File Analyzer Administration Commands
  You need `root` access to `tfactl`, or `sudo` access to run all administration commands.
- Running Oracle Trace File Analyzer Summary and Analysis Commands
  Use these commands to view the summary of deployment and status of Oracle Trace File Analyzer, and changes and events detected by Oracle Trace File Analyzer.
- Running Oracle Trace File Analyzer Diagnostic Collection Commands
  Run the diagnostic collection commands to collect diagnostic data.

E.1 Oracle Autonomous Health Framework Installation Command-Line Options

Understand the options that you can supply to the Oracle Autonomous Health Framework installer script to customize the installation.

The Oracle Autonomous Health Framework installer script:

- Takes you through an interview process if you do not specify any installation parameters.
- Appends `/oracle.ahf` to `-ahf_loc` if it does not already exist.
- Appends `/oracle.ahf/data` to `-data_dir` if it does not already exist.
- Writes the log to the `/tmp/ahf_install_`*`timestamp`*`.log` file, for example, `/tmp/ahf_install_9263_2018_09_25-07_55_52.log`.

Syntax

```
ahf_setup
[-ahf_loc AHF Location]
[-data_dir AHF Repository]
[-nodes node1,node2]
[-extract[orachk|exachk|-notfasetup]]
[-force]
[-local]
[-silent]
[-tmp_loc directory]
[-debug [-level 1-6]]
```

Parameters

Table E-1 ahf_setup Command Parameters

Parameter	Description
`-ahf_loc`	Specify the installation directory. Ensure that this directory exists before trying this option.
`-data_dir`	Specify the data directory where Oracle Autonomous Health Framework stores all the collections, metadata, and so on. Ensure that this directory exists before trying this option.

Table E-1 (Cont.) ahf_setup Command Parameters

Parameter	Description
`-nodes`	By default, Oracle Autonomous Health Framework is installed on all the cluster nodes. Specify a comma-delimited list of nodes where you want to install AHF.
`-extract`	Extracts the files from the installer. This option is default for non-root users. Specify the `-notfasetup` option just to extract and not to configure Oracle Trace File Analyzer. Run the `ahf_setup -extract [exachk|orachk]` command to install a local copy of Oracle ORAchk or Oracle EXAchk without installing the rest of AHF.
`-force`	The `-force` option is applicable only when you specify the compliance type `orachk` or `exachk` with the `-extract` option else the installer script ignores the `-force` option. `-extract orachk|exachk`
`-local`	Installs only on the local node.
`-silent`	Use this option for the Oracle Autonomous Health Framework installer script not to prompt any installation questions. If you use `-silent` option, then ensure that you use `–data_dir` option. The installer script fails if you do not use `–data_dir` option.
`-tmp_loc`	Specify a temporary location for the Oracle Autonomous Health Framework installer script to extract the install archive. Ensure that this directory exists before trying this option. Default: `/tmp`.
`-perlhome`	Specify a custom location for Perl binaries.
`-debug`	Debugs the Oracle Autonomous Health Framework installer script.
`-level`	Specify the Oracle Autonomous Health Framework Install debug level. Default `4` with option `-debug`. • `1` - FATAL • `2` - ERROR • `3` - WARNING • `4` - INFO • `5` - DEBUG • `6` - TRACE

Understanding the Location of the Data Directory

- If you install Oracle Autonomous Health Framework using the `-data_dir` option, then the installer script uses the location that you specify. The installer script will not create the specified data directory so ensure that this directory exists before trying the `-data_dir` option. You can specify a new data directory either under the current Oracle Trace File Analyzer install location or under a different directory.

- If you install Oracle Autonomous Health Framework using the `-silent` option, then ensure that you use the `-data_dir` option, otherwise, the installer script will fail.
- If you install Oracle Autonomous Health Framework without the `-data_dir` option, then the installer script will list all possible options:
  - Oracle Autonomous Health Framework installation location (`-ahf_loc`) if the free space is more than 5 GB
  - Oracle Trace File Analyzer repository if installed outside the Oracle Grid Infrastructure Home
  - Directory one level above the Oracle Grid Infrastructure Base
  - Option to enter a different directory
- If you do not use the `-silent` option and do not specify `-ahf_loc` and `-data_dir`, then the installer script displays the default options for you to confirm.

  For example:

```
# /tmp/ahf_setup -nodes node1

AHF Installation Log : /tmp/ahf_install_15992_2019_10_10-08_07_38.log

Starting Autonomous Health Framework (AHF) Installation

AHF Version: 193000 Build Date: 201910100757

Default AHF Location : /opt/oracle.ahf

Do you want to update default AHF Location ? Y|[N] :

AHF Location : /opt/oracle.ahf

Choose Data Directory from below options :

1. /u01/app [Free Space : 6742 MB]
2. Enter a different Location

Choose Option [1 - 2] : 1

AHF Data Directory : /u01/app/oracle.ahf/data

Do you want to add AHF Notification Email IDs ? [Y]|N : n

Extracting AHF to /opt/oracle.ahf
```

E.2 Running Oracle Trace File Analyzer Administration Commands

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

Table E-2   Basic tfactl commands

Command	Description
`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 access
  Use the `tfactl access` command to enable non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections.
- tfactl availability
  Use the `tfactl availability` command to enable or disable resources for Availability Score.
- tfactl blackout
  Use the `tfactl blackout` command to suppress diagnostic collections at a more granular level.
- tfactl cell
  Use the `tfactl cell` command to print or modify various storage cell configuration.
- tfactl checkupload
  Use the `tfactl checkupload` command to validate the configured upload parameters.
- tfactl diagnosetfa
  Use the `tfactl diagnosetfa` command to collect Oracle Trace File Analyzer diagnostic data from the local node to identify issues with Oracle Trace File Analyzer.
- tfactl disable
  Use the `tfactl disable` command to prevent the Oracle Trace File Analyzer daemon from restarting.
- tfactl enable
  Use the `tfactl enable` command to enable automatic restart of the Oracle Trace File Analyzer daemon after a failure or system reboot.
- tfactl floodcontrol
  Use the `tfactl floodcontrol` command to limit or stop Oracle Trace File Analyzer collecting the same events in a given frame of time.
- tfactl get
  Use the `tfactl get` command to view the details of various Oracle Trace File Analyzer configuration settings.

- tfactl getresourcelimit
  Use the `tfactl getresourcelimit` command to fetch the details of Oracle Trace File Analyzer CPU usage limitations.
- tfactl getupload
  Use the `tfactl getupload` command to fetch the details of configured upload parameters.
- tfactl host
  Use the `tfactl host` command to add hosts to, or remove hosts from the Oracle Trace File Analyzer configuration.
- tfactl print
  Use the `tfactl print` command to print information from the Berkeley DB (BDB).
- tfactl rest
  Use the `tfactl rest` command to configure REST service.
- tfactl restrictprotocol
  Use the `tfactl restrictprotocol` command to restrict certain protocols.
- tfactl sendmail
  Use the `tfactl sendmail` command to send a test email to verify SMTP configuration.
- tfactl set
  Use the `tfactl set` command to enable or disable, or modify various Oracle Trace File Analyzer functions.
- tfactl setresourcelimit
  Use the `tfactl setresourcelimit` command to restrict the CPU usage of Oracle Trace File Analyzer.
- tfactl setupload
  Use the `tfactl setupload` command to set upload parameters.
- tfactl showrepo
  Use the `tfactl showrepo` command to get the repository locations of Oracle Autonomous Health Framework components.
- tfactl start
  Use the `tfactl start` command to start the Oracle Trace File Analyzer daemon on the local node, and also to start the desired support tool.
- tfactl startahf
  Use the `tfactl startahf` command to start the scheduler for Oracle Autonomous Health Framework components.
- tfactl status
  Use the `tfactl status` command to check the run status of Oracle Trace File Analyzer.
- tfactl statusahf
  Use the `tfactl statusahf` command to check the shceduler status for Oracle Autonomous Health Framework components.
- tfactl stop
  Use the `tfactl stop` command to stop the Oracle Trace File Analyzer daemon on the local node, and also to stop the desired support tool.
- tfactl stopahf
  Use the `tfactl stopahf` command to stop the scheduler for Oracle Autonomous Health Framework components.

- tfactl syncnodes
  Use the `tfactl syncnodes` command to generate and copy Oracle Trace File Analyzer certificates to other Oracle Trace File Analyzer nodes.
- tfactl uninstall
  Use the `tfactl uninstall` command to uninstall Oracle Autonomous Health Framework.
- tfactl upload
  Use the `tfactl upload` command to upload collections or files on demand.
- tfactl unsetresourcelimit
  Use the `tfactl unsetresourcelimit` command to unset the limitations set on Oracle Trace File Analyzer CPU usage.
- tfactl unsetupload
  Use the `tfactl unsetupload` command to unset the configured upload parameters.
- tfactl version
  Use the `tfactl version` command to check the version of Oracle Autonomous Health Framework components.

E.2.1 tfactl access

Use the `tfactl access` command to enable non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections.

Non-root users can run a subset of `tfactl` commands. Running a subset of commands enables non-root users to have controlled access to Oracle Trace File Analyzer, and to run diagnostic collections. However, `root` access is still required to install and administer Oracle Trace File Analyzer. Control non-root users and groups using the `tfactl access` command. Add or remove non-root users and groups depending upon your business requirements.

> **Note:**
>
> By default, all Oracle home owners, OS DBA groups, and ASM groups are added to the Oracle Trace File Analyzer Access Manager list while installing or upgrading Oracle Trace File Analyzer.

Syntax

```
tfactl access command [options]
Commands:lsusers|add|remove|block|unblock|enable|disable|reset|removeall


tfactl access lsusers [ -local ]


tfactl access add -user user_name [ -local ]


tfactl access remove -user user_name [ -all ] [ -local ]


tfactl access block -user user_name [ -local ]


tfactl access unblock -user user_name [ -local ]


tfactl access enable [ -local ]


tfactl access disable [ -local ]


tfactl access reset


tfactl access removeall
```

Parameters

Table E-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 who 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.

Table E-3 (Cont.) tfactl access Command Parameters

Parameter	Description
`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.

Example E-1 tfactl access

To list all the Oracle Trace File Analyzer users and groups.

```
$ tfactl access lsusers

.---------------------------------.
|     TFA Users in rws1270069     |
+-----------+-----------+---------+
| User Name | User Type | Status  |
+-----------+-----------+---------+
| oradb     | USER      | Allowed |
| oragrid   | USER      | Allowed |
'-----------+-----------+---------'
```

To add a user, for example, *abc* to the Oracle Trace File Analyzer access list and enable access to Oracle Trace File Analyzer across cluster.

```
$ tfactl access add -user abc
```

To add all members of a group, for example, *xyz* to the Oracle Trace File Analyzer access list and enable access to Oracle Trace File Analyzer on the localhost.

```
$ tfactl access add -group xyz -local
```

To remove a user, for example, *abc* from the Oracle Trace File Analyzer access list.

```
$ tfactl access remove -user abc
```

To block a user, for example, *xyz* from accessing Oracle Trace File Analyzer.

```
$ tfactl access block -user xyz
```

To remove all Oracle Trace File Analyzer users and groups.

```
$ tfactl access removeall
```

E.2.2 tfactl availability

Use the `tfactl availability` command to enable or disable resources for Availability Score.

Syntax

```
tfactl enable -key key -value value  | -list


tfactl disable -key key -value value [-for nd|D|h|H|m|M]  | -list [-for nd|D|
h|H|m|M]
```

Parameters

Table E-4 tfactl enable Command Parameters

Parameter	Description
`-type resource_type`	Specify the resource type that you want to enable.
`-key key`	Specify the key of the resource that you want to enable.
`-list`	Displays the list of resources that are available for enabling.

Parameters

Table E-5 tfactl disable Command Parameters

Parameter	Description
`-type resource_type`	Specify the resource type that you want to enable.
`-key key`	Specify the key of the resource that you want to enable.
`[-for nd|D|h|H|m|M]	-list [-for nd|D|h|H|m|M]`	Specify the days, hours, or minutes to determine how long the resource will be disabled. Default is 7 days.
`-list`	Displays the list of resources that are available for disabling..

E.2.3 tfactl blackout

Use the `tfactl blackout` command to suppress diagnostic collections at a more granular level.

Syntax

```
tfactl blackout
[add|remove|print]
[-targettype all|crs|asm|asmdg|database|listener|service|os]
[-target all|name]
[-event all|"event_str1,event_str2"]
[-timeout nh|nd|none]
[-c|-local]
```

```
[-reason "reason for blackout"]
[-docollection]
```

Parameters

Table E-6 tfactl blackout Command Parameters

Parameter	Description
`add|remove|print`	Adds, removes, or prints blackout conditions.
`-targettype` *`type`*	Limits blackout only to the specified target type.
Target type: `all`|`crs`|`asm`|`asmdg`|`database`|`listener`|`service`|`os`	By default, the target type is set to `all`. `all`: The whole node is under blackout . If `all` under blackout, then every blackout element that's shown true in the Telemetry JSON will have the reason for the blackout. `crs`: Blackout the availability of the Oracle Clusterware resource or events in the Oracle Clusterware logs. `asm`**:** Blackout the availability of Oracle Automatic Storage Management (Oracle ASM) on this machine or events in the Oracle ASM alert logs. `asmdg`: Blackout an Oracle ASM disk group. `database`: Blackout the availability of an Oracle Database, Oracle Database backup, tablespace, and so on, or events in the Oracle Database alert logs. `listener`: Blackout the availability of a listener. `service`: Blackout the availability of a service. `os`: Blackout one or more operating system records.
`-target all|`*`name`*	Specify the target for blackout. You can specify a comma-delimited list of targets. By default, the target is set to `all`.
`-events all|"`*`str1`*`,`*`str2`*`"`	Limits blackout only to the availability events, or event strings, which should not trigger auto collections, or be marked as blacked out in telemetry JSON. `all`: Blackout everything for the target specified. *string*: Blackout for incidents where any part of the line contains the strings specified. Specify a comma-delimited list of strings.
`-timeout` *`nh`*`|`*`nd`*`|none`	Specify the duration for blackout in number of hours or days before timing out. By default, the timeout is set to 24 hours (24h).
`-c|-local`	Specify if blackout should be set to cluster-wide or local. By default, blackout is set to `local`.
`-reason` *`comment`*	Specify a descriptive reason for the blackout.
`-docollection`	Use this option to do an automatic diagnostic collection even if a blackout is set for this target.

Example E-2 Adding Blackout details

```
# tfactl blackout add -targettype database -target mydb -event "ORA-00600"
```

Event "ORA-00600" is blacked out until Wed Feb 20 00:20:34 PST 2019 on targettype : database, target : mydb

```
# tfactl  blackout add -targettype database -target all  -event "ORA-04031"  -
timeout 1h
```

Event "ORA-04031" is blacked out untill Tue Feb 19 01:21:27 PST 2019 on targettype : database, target : all

```
# tfactl blackout add -targettype database -target all -event "ORA-04030"  -
timeout none -c
```

Event "ORA-04030" is blacked out untill Sun Feb 19 00:22:39 PST 2119 on targettype : database, target : all

```
# tfactl blackout add -targettype all -event all -target all -timeout 1h -
reason "Disabling all events during patching"
```

Event "ALL" is blacked out untill Tue Feb 19 01:23:47 PST 2019 on targettype : all, target : all

Example E-3   Printing Blackout details

```
# tfactl blackout print
.-------------------------------------------------------------------------------
---------------------------------------------------------------------------.
| Target Type | Target | Events    | Start Time                   | End 
Time                     | Do Collection |
Reason                               |
+-------------+--------+-----------+------------------------------
+------------------------------+---------------
+--------------------------------------+
| ALL         | ALL    | ALL       | Tue Feb 19 00:23:47 PST 2019 | Tue Feb 
19 01:23:47 PST 2019 | false         | Disabling all events during patching |
| DATABASE    | ALL    | ORA-04030 | Tue Feb 19 00:22:39 PST 2019 | Sun Feb 
19 00:22:39 PST 2119 | false         | NA                                   |
| DATABASE    | ALL    | ORA-04031 | Tue Feb 19 00:21:27 PST 2019 | Tue Feb 
19 01:21:27 PST 2019 | false         | NA                                   |
| DATABASE    | MYDB   | ORA-00600 | Tue Feb 19 00:20:34 PST 2019 | Wed Feb 
20 00:20:34 PST 2019 | false         | NA                                   |
'-------------+--------+-----------+------------------------------
+------------------------------+---------------
+--------------------------------------'
```

Example E-4   Removing Blackout details

```
# tfactl blackout remove -targettype database -event "ORA-00600" -target all
```

Failed to remove Blackout details in TFA.

```
# tfactl blackout remove -targettype database -event "ORA-00600" -target mydb
```

Removed Blackouts for ORA-00600 events for targettype : database, target : mydb

```
# tfactl blackout remove -targettype database -event all -target mydb
```

Removed Blackouts for ALL events for targettype : database, target : mydb

```
# tfactl blackout remove -targettype all -event all -target all
```

Removed Blackouts for ALL events for targettype : all, target : all

E.2.4 tfactl cell

Use the `tfactl cell` command to print or modify various storage cell configuration.

Syntax

```
tfactl cell -h
```

```
tfactl cell status
```

```
tfactl cell config
```

```
tfactl cell add walletpassword
```

```
tfactl cell remove walletpassword
```

```
tfactl cell configure
```

```
tfactl cell deconfigure
```

Parameters

Table E-7 tfactl access Command Parameters

Parameter	Description
`status`	Prints the current status of storage cells.
`config`	Prints the current configuration of storage cells.
`add walletpassword`	Adds wallet or wallet password for storage cells. Oracle Trace File Analyzer uses an Oracle wallet to store cell user passwords. The wallet password is required to access the wallet to get these passwords when Oracle Trace File Analyzer runs. Running this command will store the password securely for use when collections are made. If the password is not stored, then it will have to be entered at collection time.

Table E-7 (Cont.) tfactl access Command Parameters

Parameter	Description
`remove walletpassword`	Removes wallet password. Use this command to stop Oracle Trace File Analyzer storing the wallet password.
`configure`	Configures storage cells. Used the `configure` option to configure cell collections where this was not completed at installation time, was not completed due to upgrade or following a previous deconfigure.
`deconfigure`	Removes all of the storage cell configuration.

Example E-5 tfactl cell status

```
# tfactl cell status
.----------------------------------------------------------.
| | EXADATA CELL | CURRENT STATUS |
+---+-----------------------+---------------------------+
| 1 | cel01 | ONLINE |
| 2 | cel02 | ONLINE |
'----+------------------------+---------------------------
```

Example E-6 tfactl cell config

```
# tfactl cell config
-------- ------------------------.
| Storage Cell Configuration |
+---------------------------------------------+---------------------------
--------------------------------+
| Configuration Parameter | Value |
+---------------------------------------------+----------------------------
--------------------------------+
| Exadata Support | YES |
| Configured Storage Cells | YES |
| Oracle Wallet Used | YES |
| Oracle Wallet Location | /u01/app/tfa/db01/tfa_home/internal/tfawallet |
| Oracle Wallet Password is with TFA | YES |
| Oracle Wallet Password Storage Status | Stored |
'---------------------------------------+-------------------------------------
```

Example E-7 tfactl cell add walletpassword

```
# tfactl cell add walletpassword
Please Enter Password for Oracle Wallet:
```

Example E-8 tfactl cell remove walletpassword

```
# tfactl cell remove walletpassword
Please Enter Password for Oracle Wallet:
Oracle Wallet Password is successfully removed.
```

Example E-9 tfactl cell deconfigure

```
# tfactl cell deconfigure
Removing Storage Cell Configuration...
Successfully removed Storage Cell Configuration.
```

E.2.5 tfactl checkupload

Use the `tfactl checkupload` command to validate the configured upload parameters.

You can run the `checkupload` command as `root` or a non-root user.

Syntax

```
tfactl checkupload
[-h][--help]
[-name NAME]
```

Parameters

Table E-8 tfactl checkupload Command Parameters

Parameter	Description
`name`	Specify the name of your configuration. For example, *mosconfig* to upload to My Oracle Support.

E.2.6 tfactl diagnosetfa

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

Syntax

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

Parameters

Table E-9 tfactl diagnosetfa Command Parameters

Parameter	Description
`-repo` *`repository`*	Specify the repository directory for the 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.

E.2.7 tfactl disable

Use the `tfactl disable` command to prevent the Oracle Trace File Analyzer daemon from restarting.

Syntax

```
tfactl disable
```

E.2.8 tfactl enable

Use the `tfactl enable` command to enable automatic restart of the Oracle Trace File Analyzer daemon after a failure or system reboot.

Syntax

```
tfactl enable
```

E.2.9 tfactl floodcontrol

Use the `tfactl floodcontrol` command to limit or stop Oracle Trace File Analyzer collecting the same events in a given frame of time.

Syntax

```
tfactl floodcontrol
[-h][--help]
print|update|clear
[-event name]
[-limit n]
[-limittime n]
[-pausetime n]
```

Parameters

Table E-10 tfactl floodcontrol Command Parameters

Parameter	Description
`print|update|clear`	Print, update, or clear flood control details.
`event` *name*	Flood control event name.
`limit` *n*	Flood control limit count.
`limittime` *n*	Flood control initital limit time in minutes.
`pausetime` *n*	Flood control pause time in minutes.

E.2.10 tfactl get

Use the `tfactl get` command to view the details of various Oracle Trace File Analyzer configuration settings.

Syntax

```
tfactl get
| autodiagcollect
| trimfiles
| tracelevel=COLLECT|SCAN|INVENTORY|OTHER|ISA|HANDLER|MAIN|CLIENT|CONSOLE:
FATAL|ERROR|WARNING|INFO|DEBUG|TRACE
| reposizeMB
| repositorydir
| logsize
| logcount
| maxcorefilesize
| maxcorecollectionsize
| maxfilecollectionsize
| autopurge
| publicip
| redact
| minSpaceForRTScan
| rtscan
| diskUsageMon
| diskUsageMonInterval
| manageLogsAutoPurge
| manageLogsAutoPurgeInterval
| manageLogsAutoPurgePolicyAge
| minfileagetopurge
| tfaIpsPoolSize
| tfaDbUtlPurgeAge
| tfaDbUtlPurgeMode
| tfaDbUtlPurgeThreadDelay
| tfaDbUtlCrsProfileDelay
| indexRecoveryMode
| collection.isa
| discovery
| inventory
| unreachableNodeSleepTime
| unreachableNodeTimeOut
| ipsAlertlogTrimsizeMB
| clustereventmonitor
[-node]
[-match pattern]
```

Example E-10   tfactl get

```
# tfactl get collect -match
.--------------------------------------------------------------------------.
|                                testserver                                |
+------------------------------------------------------------------+-------+
| Configuration Parameter                                          | Value |
+------------------------------------------------------------------+-------+
| collectAllDirsByFile                                             | ON    |
| Auto Diagcollection ( autodiagcollect )                          | ON    |
| ISA Data Gathering ( collection.isa )                            | ON    |
| collectTrm                                                       | OFF   |
| Generation of Mini Collections ( minicollection )                | ON    |
| chaautocollect                                                   | ON    |
| Max File Collection Size (MB) ( maxFileCollectionSize )          | 5120  |
| Max Collection Size of Core Files (MB) ( maxCoreCollectionSize ) | 500   |
| minTimeForAutoDiagCollection                                      | 12    |
'-------------------------------------------------------------------+-------'


tfactl get maxcorefilesize
.--------------------------------------------------------.
|                        testserver                      |
+------------------------------------------------+-------+
| Configuration Parameter                        | Value |
+------------------------------------------------+-------+
| Max Size of Core File (MB) ( maxCoreFileSize ) | 50    |
'------------------------------------------------+-------'


tfactl get maxcorecollectionsize
.--------------------------------------------------------------------------.
|                                 testserver                               |
+------------------------------------------------------------------+-------+
| Configuration Parameter                                          | Value |
+------------------------------------------------------------------+-------+
| Max Collection Size of Core Files (MB) ( maxCoreCollectionSize ) | 500   |
'------------------------------------------------------------------+-------'
```

E.2.11 tfactl getresourcelimit

Use the `tfactl getresourcelimit` command to fetch the details of Oracle Trace File Analyzer CPU usage limitations.

Syntax

```
tfactl getresourcelimit
[-tool tool_name]
[-resource resource_type]
```

Parameters

Table E-11 tfactl getresourcelimit Command Parameters

Parameter	Description
`tool`	Currently, you can only specify `tfa`.
`resource`	Currently, you can only specify `cpu`.

Example E-11 getresourcelimit Example

```
# tfactl getresourcelimit
Tool TFA: Resource CPU: Limit value: 1
```

E.2.12 tfactl getupload

Use the `tfactl getupload` command to fetch the details of configured upload parameters.

You can run the `getupload` command as `root` or a non-root user.

Syntax

```
tfactl getupload
[-h][--help]
[-all]
[-name NAME]
[-user USER]
[-password]
[-server SERVER]
[-url URL]
[-proxy PROXY]
[-noauth NOAUTH]
[-request REQUEST]
[-https_token HTTPS_TOKEN]
[-header HEADER]
[-secure SECURE]
[-connectstring CONNECTSTRING]
[-uploadtable UPLOADTABLE]
```

Parameters

Table E-12 tfactl getupload Command Parameters

Parameter	Description
`all`	All of the parameters.
`name`	Specify the name of your configuration. For example, *mosconfig*.
`user`	Specify the user who has the privileges to access the endpoint. For example, *upload.user@example.com*.
`password`	Specify the password of the user.
`server`	Specify the name of the server to which you have uploaded files. For example, *bugsftp.example.com*.
`url`	Specify the target URL in case of HTTPS type. For example, *https://samplehost.com*.
`proxy`	Specify the URL of the proxy server. For example, *www.example.com:80*.
`noauth`	Specify `true` and `false`. Default value is `false`. If `noauth` is set to `true`, then HTTPS upload will skip authentication. For example, upload files to PAR, Pre Authenticated URL where no user/password authentication is required.
`request`	Specify the request type, for example, `POST`.

Table E-12 (Cont.) tfactl getupload Command Parameters

Parameter	Description
`https_token`	Specify any static header values while configuring. For example, set auth tokens while configuring the HTTPS end point. You can also pass dynamic headers at upload time by passing the –`https_token` *`headers`* command option to `tfactl upload` command. For example: *`-H 'X-TFA-REQUESTID: 1'`*.
`header`	Stores the `executionId` in the `ahf.properties` file. For example, to set the header:`tfactl setupload -name a1 -type https -header X-TFA-HEADERS:executionId=aeldb1db01_2020.06.16_19.20.55.15336025`
`secure`	Specify `true` or `false`. Default value is `true`. Specifying the secure value checks for certificates. If `secure` is set to `false`, the `upload` command will run an unsecure upload.
`connectstring`	Specify the database connect string to log in to the database where you have uploaded files. For example, `(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME = orcl)))`.
`uploadtable`	Specify the name of the table where you have uploaded files as `BLOB` type. For example, for uploading Oracle ORAchk collections to the Collection Manager it is set to `RCA13_DOCS`.

E.2.13 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]
```

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 DB (BDB).

Example E-12 tfactl host

Specify a host name to add:

```
$ tfactl host add myhost
```

Specify a host name to remove:

```
$ tfactl host remove myhost
```

E.2.14 tfactl print

Use the `tfactl print` command to print information from the Berkeley DB (BDB).

Syntax

```
tfactl print command [options]
Commands:status|components|config|directories|hosts|actions|repository|
suspendedips|protocols|smtp
```

```
tfactl print status
```

```
tfactl print components [ [component_name1] [component_name2] ...
[component_nameN] ]
```

```
tfactl print config [ -node all | local | n1,n2,...  -name param]
```

```
tfactl print directories [ -node all | local | n1,n2,... ] [ -comp
component_name1,component_name2,... ] [ -policy exclusions | noexclusions ]
[ -permission public | private ]
```

```
tfactl print hosts
```

```
tfactl print actions [ -status status ] [ -since nh|d ]
```

```
tfactl print repository
```

```
tfactl print suspendedips
```

```
tfactl print protocols
```

```
tfactl print smtp
```

Parameters

Table E-13   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.
`components`	Displays the desired components in the configuration.
`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.

Table E-13 (Cont.) tfactl print Command Parameters

Parameter	Description
`hosts`	Lists the hosts that are part of the Oracle Trace File Analyzer cluster, and that can receive cluster-wide commands.
`actions`	Lists all the actions submitted to Oracle Trace File Analyzer, such as diagnostic collection. By default, `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.
`suspendedips`	Lists all paused Oracle Trace File Analyzer IPS collections.
`protocols`	Lists all available and restricted protocols.
`smtp`	Displays the SMTP server configuration

Options

Option	Description
`-status` *`status`*	Action status can be one or more of `COMPLETE`, `RUNNING`, `FAILED`, `REQUESTED` Specify a comma-separated list of statuses.
`-since` *`nh|d`*	Specify actions from past *n* days or *n* hours.

Example E-13 tfactl print smtp

```
tfactl print smtp

.---------------------------.
| SMTP Server Configuration |
+---------------+-----------+
| Parameter     | Value     |
+---------------+-----------+
| smtp.auth     | false     |
| smtp.from     | tfa       |
| smtp.user     | -         |
| smtp.cc       | -         |
| smtp.port     |        25 |
| smtp.bcc      | -         |
| smtp.password | *******   |
| smtp.host     | localhost |
| smtp.to       | -         |
| smtp.debug    | true      |
| smtp.ssl      | false     |
'---------------+-----------'
```

Example E-14 tfactl print protocols

```
tfactl print protocols

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

Example E-15 tfactl print components ASM

```
$ tfactl print components ASM

.--------------------------------------------------.
|                  XML Components                  |
+---------------+----------------------------------+
| Field         | Value                            |
+---------------+----------------------------------+
| Name          | ASM                              |
| Description   | ASM logs                         |
| Comp. Types   | collection action                |
| Configuration | all                              |
| Subcomponents | name:instance required: default: |
| Also collect  | TNS                              |
|               | AFD                              |
|               | ASMPROXY                         |
|               | ASMIO                            |
'---------------+----------------------------------'
```

Example E-16 tfactl print components ODASTORAGE

```
$ tfactl print components ODASTORAGE

.-------------------------------------------.
|               XML Components              |
+---------------+---------------------------+
| Field         | Value                     |
+---------------+---------------------------+
| Name          | ODASTORAGE                |
| Description   | ODA Storage logs and Data |
| Comp. Types   | action                    |
| Configuration | ODA                       |
| Also collect  | OS                        |
|               | ODA                       |
|               | ASM                       |
|               | DCS                       |
'---------------+---------------------------'
```

Example E-17 tfactl print config

```
tfactl print config
.------------------------------------------------------------------------------
-------.
|
node1                                     |
+-----------------------------------------------------------------------
+------------+
| Configuration Parameter                                               |
Value      |
+-----------------------------------------------------------------------
+------------+
| TFA Version                                                           |
19.1.0.0.0 |
| Java Version                                                          |
1.8        |
| Public IP Network                                                     |
false      |
| 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          |
| Granular Tracing                                                      |
false      |
| Debug Mask (Hex)                                                      |
0          |
| Repository current size (MB)                                          |
146        |
| Repository maximum size (MB)                                          |
10240      |
| Max Size of TFA Log (MB)                                              |
50         |
| Max Number of TFA Logs                                                |
10         |
| Max Size of Core File (MB)                                            |
50         |
| Max Collection Size of Core Files (MB)                                |
500        |
| Max File Collection Size (MB)                                         |
5120       |
| 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          |
| TFA ISA Purge Age (seconds)                                           |
604800     |
| TFA ISA Purge Mode                                                    |
profile    |
| TFA ISA Purge Thread Delay (minutes)                                  |
60         |
| Setting for ACR redaction (none|SANITIZE|MASK)                        |
none       |
| Email Notification will be sent for CHA EVENTS if address is set      |
false      |
| AUTO Collection will be generated for CHA EVENTS                      |
false      |
'-----------------------------------------------------------------------
+------------'
```

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.

E.2.15 tfactl rest

Use the `tfactl rest` command to configure REST service.

Syntax

```
tfactl rest
[-status|-start|-stop|-upgrade|-uninstall]
[-dir directory]
[-port port]
[-user user]
[-debug [-level debug_level 1-6]]
```

> **Note:**
>
> You can run the REST command only as `root` user.

Parameters

Table E-14 REST Command Parameters

Parameter	Description
`-status`	Prints the current status.
`-start`	Starts Oracle Trace File Analyzer REST services if not already running.
`-stop`	Stops Oracle Trace File Analyzer REST services if running.
`-upgrade`	Checks if the configured ORDS API should be upgraded. If the ORDS API needs upgrading, then stops ORDS, upgrades the API, and then restarts ORDS.
`-uninstall`	Removes the Oracle Trace File Analyzer REST configuration.
`-dir`	The directory to use to store the Oracle Trace File Analyzer REST configuration details. Defaults to the users home directory.
`-port`	The port to run ORDS on. Defaults to 9090.
`-user`	The user to start ORDS as. Defaults to the GRID owner.
`-debug`	Enables debug.

Table E-14 (Cont.) REST Command Parameters

Parameter	Description
`-level`	The level of debug to use, where available levels are: • 1 – FATAL • 2 – ERROR • 3 – WARNING • 4 – INFO (default) • 5 – DEBUG • 6 – TRACE

E.2.16 tfactl restrictprotocol

Use the `tfactl restrictprotocol` command to restrict certain protocols.

Syntax

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

Example E-18 tfactl restrictprotocol

```
$ tfactl restrictprotocol TLSv1
```

E.2.17 tfactl sendmail

Use the `tfactl sendmail` command to send a test email to verify SMTP configuration.

Syntax

```
tfactl sendmail email_address
```

E.2.18 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
| trimfiles=ON|OFF
| tracelevel=COLLECT|SCAN|INVENTORY|OTHER|ISA|HANDLER|MAIN|CLIENT|CONSOLE:
FATAL|ERROR|WARNING|INFO|DEBUG|TRACE
| reposizeMB=n
| repositorydir=dir [-force]
| logsize=n [-local]
| logcount=n [-local]
| port=n
| maxcorefilesize=n [-local]
| maxcorecollectionsize=n [-local]
| maxfilecollectionsize=n
| autopurge=ON|OFF
| publicip=ON|OFF
| smtp
| minSpaceForRTScan=n
| rtscan=ON|OFF
| diskUsageMon=ON|OFF
| diskUsageMonInterval=n
| manageLogsAutoPurge=ON|OFF
| manageLogsAutoPurgeInterval=n
| manageLogsAutoPurgePolicyAge=d|h
| minagetopurge=n
| tfaDbUtlPurgeAge=n
| tfaDbUtlPurgeMode=simple|resource|profile
| tfaDbUtlPurgeThreadDelay
| tfaDbUtlCrsProfileDelay
| indexRecoveryMode
| rediscoveryInterval]
[-c]
[-local]
```

Parameters

Table E-15 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.
`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|ISA|HANDLER|MAIN|CLIENT|CONSOLE: FATAL|ERROR|WARNING|INFO|DEBUG|TRACE`	Controls the trace level of log files. **Note:** Do not change the tracing level unless you are directed to do so by My Oracle Support.
`reposizeMB=`*`number`*	Sets the maximum size in MB of the collection repository.
`repositorydir=`*`directory`* `[-force]`	Specify the collection repository directory. Use the `-force` option to skip initial checks while changing the repository (Not Recommended)

Table E-15 (Cont.) tfactl set Command Parameters

Parameter	Description
`logsize=n [-local]`	Sets the maximum size, in MB, of each log before Oracle Trace File Analyzer rotates to a new log. • **Default:** 50 MB • **Minimum:** 10 MB • **Maximum:** 500 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:** 10 • **Minimum:** 5 • **Maximum:** 50 Use the `-local` option to apply the change only to the local node.
`port=n`	Specify the Oracle Trace File Analyzer port.
`maxcorefilesize=n [-local]`	Sets the maximum size of the core files to the size specified in MB. **Default:** 50 MB
`maxcorecollectionsize=n`	Sets the maximum collection size of the core files to the size specified in MB. **Default:** 500 MB
`maxfilecollectionsize=n`	Specify the file size in MB (5 GB by default). When you run the `tfactl diagcollect` command, it adds only the last 200 KB of the files that exceed the maximum file size to the diagnostic collection. The `tfactl diagcollect` command adds a new file, `skipped_files.txt` with the list of skipped files that are too large to add to the diagnostic collection.
`autopurge=ON|OFF`	When set to `ON`, enables automatic purging of collections when Oracle Trace File Analyzer observes less space in the repository (`ON` by default).
`publicip=ON|OFF`	Allows Oracle Trace File Analyzer to run on public network.
`smtp`	Specify the configuration details for the SMTP server to use for email notifications when prompted.
`minSpaceForRTScan=n`	Specify the minimum space required to run RT scan (500 by default).
`rtscan`	Specify to allow Oracle Trace File Analyzer to perform alert log scanning.
`diskUsageMon=ON|OFF`	Turns `ON` or `OFF` monitoring disk usage and recording snapshots (`ON` by default). Oracle Trace File Analyzer stores the snapshots under `tfa/repository/suptools/node/managelogs/usage_snapshot/`.
`diskUsageMonInterval=minutes`	Specify the time interval between snapshots. **Default:** 60 minutes
`manageLogsAutoPurge=ON | OFF`	Turns automatic purging on or off (`ON` by default in DSC and `OFF` by default elsewhere).
`manageLogsAutoPurgeInterval=minutes`	Specify the purge frequency. **Default:** 60 minutes
`manageLogsAutoPurgePolicyAge=nd|h`	Age of logs to be purged. **Default:** 30 days

Table E-15 (Cont.) tfactl set Command Parameters

Parameter	Description
`minagetopurge=n`	Set the minimum age, in hours, for a collection before Oracle Trace File Analyzer considers it for purging. - **Default:** 12 hours - **Minimum:** 12 hours - **Maximum:** 168 hours
`tfaDbUtlPurgeAge=n`	Sets the Oracle Trace File Analyzer ISA purge age in seconds. **Default:** 604800 seconds, that is, 7 days **Range:** 86400 (1 day) - 2592000 (1 month)
`tfaDbUtlPurgeMode=simple|resource|profile`	Sets the Oracle Trace File Analyzer ISA purge mode.
`tfaDbUtlPurgeThreadDelay=n`	Set the Oracle Trace Fils Analyzer ISA purge thread delay in minutes. **Default:** 60 minutes **Range:** 1 - 1440 (24 hours) minutes
`tfaDbUtlCrsProfileDelay=n`	Set the Oracle Trace File Analyzer ISA CRS profile delay in minutes. **Default:** 30 minutes **Range:** 1 - 60 minutes
`indexRecoveryMode=recreate|restore`	Set the Lucene index recovery mode to recreate or restore. **Recreate:** If there's corruption, then index will be recreated with no recovery. **Restore:** If there's corruption, then index will be recovered from last backup and the latest changes are reapplied
`rediscoveryInterval`	Sets the time interval for running lite rediscovery. **Minimum:** 10 minutes **Maximum:** 1 day
`-c`	Propagates the settings to all nodes in the Oracle Trace File Analyzer configuration.
`-local`	Set the value on the local node. If the option is not included, then the value will be set on all the nodes.

Example E-19 tfactl set

```
$ tfactl set autodiagcollect=ON reposizeMB=20480
$ tfactl set autodiagcollect=ON
$ tfactl set autopurge=ON
$ tfactl set tracelevel=INVENTORY:DEBUG
$ tfactl set reposizeMB=20480
$ tfactl set logsize=100
$ tfactl set port=5000
```

Example E-20 tfactl set rediscoveryInterval

```
tfa/bin/tfactl set rediscoveryInterval=1m1h1d
Successfully set rediscoveryInterval=1m1h1d
.-------------------------------------------------------.
|                         node1                          |
+----------------------------------------------+--------+
| Configuration Parameter                      | Value  |
+----------------------------------------------+--------+
| Rediscovery Interval ( rediscoveryInterval ) | 1m1h1d |
'----------------------------------------------+--------'
```

E.2.19 tfactl setresourcelimit

Use the `tfactl setresourcelimit` command to restrict the CPU usage of Oracle Trace File Analyzer.

Syntax

```
tfactl setresourcelimit
[-tool tool_name]
[-resource resource_type]
[-value value]
```

Parameters

Table E-16 tfactl setresourcelimit Command Parameters

Parameter	Description
`value`	Set the limit to a minimum of 50% of a single CPU, and a maximum of 4 or 75% of the available CPUs, whichever is lower. By default, the CPU limit is set to the maximum. To limit TFA to a maximum of 50% of a single CPU: `tfactl setresourcelimit -value 0.5`
`tool`	Currently, you can only specify `tfa`. Default: *`tfa`*
`resource`	Currently, you can only specify `cpu`. Default: *`cpu`*

Example E-21 setresourcelimit Examples

On a server with 10 CPUs, the default limit will be 4 CPUs:

```
# tfactl setresourcelimit
Tool TFA: Resource CPU: Limit value: 4
```

On a server with 4 CPUs, the default limit will be 3 CPUs (75% of available CPUs):

```
tfactl setresourcelimit
Tool TFA: Resource CPU: Limit value: 3


# tfactl setresourcelimit -value 2
Tool TFA: Resource CPU: Limit value: 2
```

E.2.20 tfactl setupload

Use the `tfactl setupload` command to set upload parameters.

You can run the `setupload` command as `root` or a non-root user.

Syntax

```
tfactl setupload
[-h][--help]
[-all]
[-type TYPE]
[-name NAME]
[-user USER]
[-password]
[-server SERVER]
[-url URL]
[-proxy PROXY]
[-noauth NOAUTH]
[-https_token HTTPS_TOKEN]
[-request REQUEST]
[-header HEADER]
[-secure SECURE]
[-connectstring CONNECTSTRING]
[-uploadtable UPLOADTABLE]
```

Parameters

Table E-17 tfactl setupload Command Parameters

Parameter	Description
`all`	All of the parameters.
`type`	Specify the type of an endpoint. For example, `https`, `sftp`, or `sqlnet`.
`name`	Specify a unique descriptive name to your configuration. For example, *mosconfig* to upload to My Oracle Support.
`user`	Specify the user who has the privileges to access the endpoint. For example, *upload.user@example.com*.
`password`	Specify the password of the user.
`server`	Specify the name of the server to which you want to upload files. For example, *bugsftp.example.com*.
`url`	Specify the target URL to upload files in case of HTTPS type. For example, *https://samplehost.com*.
`proxy`	Specify the URL of the proxy server. For example, *www.example.com:80*.
`noauth`	Specify `true` and `false`. Default value is `false`. If `noauth` is set to `true`, then HTTPS upload will skip authentication. For example, upload files to PAR, Pre Authenticated URL where no user/password authentication is required.
`request`	Specify the request type, for example, `POST`.

Table E-17 (Cont.) tfactl setupload Command Parameters

Parameter	Description
`https_token`	Specify any static header values while configuring. For example, set auth tokens while configuring the HTTPS end point. For example, `tfactl setupload -name config -type https -https_token 'abc:13'`. You can also pass dynamic headers at upload time by passing the `-https_token` *`headers`* command option to `tfactl upload` command. For example: *`-H 'X-TFA-REQUESTID: 1'`*.
`header`	Stores the `executionId` in the `ahf.properties` file. For example, to set the header:`tfactl setupload -name a1 -type https -header X-TFA-HEADERS:executionId=aeldb1db01_2020.06.16_19.20.55.15336025`
`secure`	Specify `true` or `false`. Default value is `true`. Specifying the secure value checks for certificates. If `secure` is set to `false`, then the `upload` command will run an unsecure upload.
`connectstring`	Specify the database connect string to log in to the database where you want to upload files. For example, `(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME = orcl)))`.
`uploadtable`	Specify the name of the table where you want to upload files as `BLOB` type. For example, for uploading Oracle ORAchk collections to the Collection Manager it is set to `RCA13_DOCS`.

To setup MOS configuration:

```
tfactl setupload -name mos -type https -user sample_user@domain.com -url
https://transport.oracle.com/upload/issue
```

To set proxy for MOS configuration:

```
tfactl setupload -name mos -type https -proxy www-proxy.server.com:80
```

To upload to MOS using `tfactl upload`:

```
tfactl upload -name mos -id 3-23104325631 -file /opt/oracle.ahf/data/
repository/auto_srdc_ORA-00600_20200706T18:58:09_myserver1.zip
```

To upload to MOS using `tfactl diagcollect`:

```
tfactl diagcollect -upload mos -srdc ORA-00600 -id 3-23104325631
```

or

```
tfactl diagcollect -srdc ORA-00600 -sr 3-23104325631
```

> **Note:**
>
> Ensure that the configuration name is `mos`.

E.2.21 tfactl showrepo

Use the `tfactl showrepo` command to get the repository locations of Oracle Autonomous Health Framework components.

Syntax

```
tfactl showrepo
[-h]
[-all]
[-tfa]
[-compliance]
```

Parameters

Table E-18   tfactl showrepo Command Parameters

Parameter	Description
`-all`	Displays the repository locations of Oracle Autonomous Health Framework components.
`-tfa`	Displays the repository locations of Oracle Trace File Analyzer.
`-compliance`	Displays the repository locations of Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components.

E.2.22 tfactl start

Use the `tfactl start` command to start the Oracle Trace File Analyzer daemon on the local node, and also to start the desired support tool.

Syntax

```
tfactl start [tool]
```

E.2.23 tfactl startahf

Use the `tfactl startahf` command to start the scheduler for Oracle Autonomous Health Framework components.

Syntax

```
tfactl startahf
[-h]
[-all]
[-tfa tfa_start_args]
[-compliance compliance_autostart_args]
```

Parameters

Table E-19 tfactl startahf Command Parameters

Parameter	Description
`-all`	Starts the Oracle Trace File Analyzer and Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.
`-tfa`	Starts the Oracle Trace File Analyzer daemon.
`-tfa` *`tfa_start_args`*	Starts the Oracle Trace File Analyzer daemon with the option specified. You can specify all Oracle Trace File Analyzer supported options. For example: `tfactl startahf -tfa "tfa_start_args"`
`-compliance`	Starts the Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.

`-compliance` *`compliance_autostart_args`*	Starts the Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons with the option specified. Prepend the argument with a space followed by an hyphen and then wrap it with double quotes. You can specify all Oracle ORAchk and Oracle EXAchk supported options. For example:
	`tfactl startahf -compliance " -compliance_autostart_args"`
	`tfactl startahf -compliance -cargs " -c X4-2,EXAMAA"`
	`tfactl startahf -compliance -cargs " -debug"`
	`tfactl startahf -compliance -cagrs " -withisa"`

E.2.24 tfactl status

Use the `tfactl status` command to check the run status of Oracle Trace File Analyzer.

Syntax

```
tfactl status
```

E.2.25 tfactl statusahf

Use the `tfactl statusahf` command to check the shceduler status for Oracle Autonomous Health Framework components.

Syntax

```
tfactl statusahf [-h]
[-all]
[-tfa]
[-compliance]
```

Parameters

Table E-20 tfactl statusahf Command Parameters

Parameter	Description
`-all`	Checks and displays the status of Oracle Trace File Analyzer and Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.
`-tfa`	Checks and displays the status of Oracle Trace File Analyzer daemon.
`-compliance`	Checks and displays the status of Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.

E.2.26 tfactl stop

Use the `tfactl stop` command to stop the Oracle Trace File Analyzer daemon on the local node, and also to stop the desired support tool.

Syntax

```
tfactl stop [tool]
```

E.2.27 tfactl stopahf

Use the `tfactl stopahf` command to stop the scheduler for Oracle Autonomous Health Framework components.

Syntax

```
tfactl stopahf [-h]
[-all]
[-tfa]
[-compliance]
```

Parameters

Table E-21 tfactl stopahf Command Parameters

Parameter	Description
`-all`	Stops the Oracle Trace File Analyzer and Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.
`-tfa`	Stops the Oracle Trace File Analyzer daemon.
`-compliance`	Stops the the Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components daemons.

E.2.28 tfactl syncnodes

Use the `tfactl syncnodes` command to generate and copy Oracle Trace File Analyzer certificates to other Oracle Trace File Analyzer nodes.

Syntax

```
tfactl syncnodes [-regenerate]
```

Parameters

Table E-22 tfactl syncnodes Command Parameters

Parameter	Description
`-regenerate`	Regenerates Oracle Trace File Analyzer certificates.

E.2.29 tfactl uninstall

Use the `tfactl uninstall` command to uninstall Oracle Autonomous Health Framework.

Syntax

Run the `uninstall` command as `root`, or install user

```
tfactl uninstall
```

E.2.30 tfactl upload

Use the `tfactl upload` command to upload collections or files on demand.

You can run the `upload` command as `root` or a non-root user.

Syntax

```
tfactl upload
[-sr sr_number]
[-name config_name]
```

```
[-id the location or target where you want to upload your files to]
[-file file_name]
```

Parameters

Table E-23 tfactl upload Command Parameters

Parameter	Description
`-sr sr_number`	Specify the SR number.
`-name config_name`	Specify a unique name for the configuration.
`-id The location or target where you want to upload your files to.`	Specify the location or target where you want to upload your files to.
`-file file_name`	Specify the name of the file to upload.

Example E-22 Upload to MOS using tfactl upload Example

```
tfactl upload -name mos -id 3-23104325631 -file /opt/oracle.ahf/data/
repository/auto_srdc_ORA-00600_20200706T18:58:09_myserver1.zip
```

Example E-23 Upload to MOS using tfactl diagcollect Example

```
tfactl diagcollect -upload mos -srdc ORA-00600 -id 3-23104325631


tfactl diagcollect -srdc ORA-00600 -sr 3-23104325631
```

> **Note:**
>
> Ensure that the configuration name is `mos`.

For more information on configuration setup, run `tfactl setupload -h`.

E.2.31 tfactl unsetresourcelimit

Use the `tfactl unsetresourcelimit` command to unset the limitations set on Oracle Trace File Analyzer CPU usage.

Syntax

```
tfactl unsetresourcelimit
[-tool tool_name]
[-resource resource_type]
```

Parameters

Table E-24 tfactl unsetresourcelimit Command Parameters

Parameter	Description
`tool`	Currently, you can only specify `tfa`.
`resource`	Currently, you can only specify `cpu`.

Example E-24 unsetresourcelimit Example

```
# tfactl unsetresourcelimit -tool tfa -resource cpu
```

E.2.32 tfactl unsetupload

Use the `tfactl unsetupload` command to unset the configured upload parameters.

You can run the `unsetupload` command as `root` or a non-root user.

Syntax

```
tfactl unsetupload
[-h][--help]
[-all]
[-name NAME]
[-user USER]
[-password]
[-server SERVER]
[-url URL]
[-proxy PROXY]
[-noauth NOAUTH]
[-https_token HTTPS_TOKEN]
[-request REQUEST]
[-header HEADER]
[-secure SECURE]
[-connectstring CONNECTSTRING]
[-uploadtable UPLOADTABLE]
```

Parameters

Table E-25 tfactl unsetupload Command Parameters

Parameter	Description
`all`	All of the parameters.
`name`	Specify the name of your configuration. For example, *mosconfig* to upload to My Oracle Support.
`user`	Specify the user who has the privileges to access the endpoint. For example, *upload.user@example.com*.
`password`	Specify the password of the user.
`server`	Specify the name of the server to which you have uploaded the files. For example, *bugsftp.example.com*.

Table E-25 (Cont.) tfactl unsetupload Command Parameters

Parameter	Description
`url`	Specify the target URL to which you have uploaded the files in case of HTTPS type. For example, *https://samplehost.com*.
`proxy`	Specify the URL of the proxy server. For example, *www.example.com:80*.
`noauth`	Specify `true` and `false`. Default value is `false`. If `noauth` is set to `true`, then HTTPS upload will skip authentication. For example, upload files to PAR, Pre Authenticated URL where no user/password authentication is required.
`request`	Specify the request type, for example, `POST`.
`https_token`	Specify any static header values while configuring. For example, set auth tokens while configuring the HTTPS end point. You can also pass dynamic headers at upload time by passing the `-https_token` *`headers`* command option to `tfactl upload` command. For example: *`-H 'X-TFA-REQUESTID: 1'`*.
`header`	Stores the `executionId` in the `ahf.properties` file. For example, to set the header:`tfactl setupload -name a1 -type https -header X-TFA-HEADERS:executionId=aeldb1db01_2020.06.16_19.20.55.15336025`
`secure`	Specify `true` or `false`. Default value is `true`. Specifying the secure value checks for certificates. If `secure` is set to `false`, then the `upload` command will run an unsecure upload.
`connectstring`	Specify the database connect string to log in to the database where you have uploaded files. For example, `(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST = host)(PORT = 1521))(CONNECT_DATA =(SERVER = DEDICATED)(SERVICE_NAME = orcl)))`.
`uploadtable`	Specify the name of the table where you have uploaded the files as `BLOB` type. For example, for uploading Oracle ORAchk collections to the Collection Manager it is set to `RCA13_DOCS`.

E.2.33 tfactl version

Use the `tfactl version` command to check the version of Oracle Autonomous Health Framework components.

Syntax

```
tfactl version
[-h]
[-all]
[-tfa]
[-compliance]
```

Parameters

Table E-26 tfactl version Command Parameters

Parameter	Description
`-all`	Checks and displays the version of Oracle Autonomous Health Framework components.
`-tfa`	Checks and displays the version of Oracle Trace File Analyzer.
`-compliance`	Checks and displays the version of Oracle Autonomous Health Framework compliance (Oracle ORAchk and Oracle EXAchk) components.

E.3 Running Oracle Trace File Analyzer Summary and Analysis Commands

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

- tfactl analyze
  Use the `tfactl analyze` command to obtain analysis of your system by parsing the database, Oracle Automatic Storage Management (Oracle ASM), and Oracle Grid Infrastructure alert logs, system message logs, OSWatcher Top, and OSWatcher Slabinfo files.
- tfactl changes
  Use the `tfactl changes` command to view the changes detected by Oracle Trace File Analyzer.
- tfactl events
  Use the `tfactl events` command to view the events detected by Oracle Trace File Analyzer.
- tfactl isa
  Use the `tfactl isa` command to view the Infrastructure Service Automation (ISA) score.
- tfactl run
  Use the `tfactl run` command to run the requested action (can be inventory or scan or any support tool).
- tfactl search
  Use the `tfactl search` command to search all metadata stored in the Oracle Trace File Analyzer index.
- tfactl summary
  Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.
- tfactl toolstatus
  Use the `tfactl toolstatus` command to view the status of Oracle Trace File Analyzer Support Tools across all nodes.

E.3.1 tfactl analyze

Use the `tfactl analyze` command to obtain analysis of your system by parsing the database, Oracle Automatic Storage Management (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|oratop|all]
[-type error|warning|generic]
[-last n[h|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]
[-examples]
```

Parameters

Table E-27 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|oratop|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`.
`[-last n[h|d]]`	Specify an amount of time, in hours or days, before current time that you want Oracle Trace File Analyzer to analyze.
`-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.
`[-examples]`	Specify this parameter to view `analyze` usage examples.

-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 E-28 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.

oratop options

The options available when using `-comp oratop`:

```
-database dbname oratop options logon
```

Table E-29 tfactl analyze -comp oratop options

Argument	Description
`-database` *`dbname`*	Specify the name of the Oracle Database to run `oratop`.

Table E-29 (Cont.) tfactl analyze -comp oratop options

Argument	Description
`logon`	Default is `/ as sysdba`. Specify a different user using, `{username[/password][@connect_identifier] | / } [AS {SYSDBA|SYSOPER}]` Connect Identifier: *`host`*`[:`*`port`*`]/[`*`service_name`*`]`

Table E-30 oratop options

Argument	Description
`-d`	Real-time (RT) wait events (section 3. Default is Cumulative..
`-k`	FILE#:BLOCK#, section 4 lt is (EVENT/LATCH).
`-m`	Specify `MODULE`/`ACTION` (section 4). Default is `USERNAME`/`PROGRAM`.
`-s`	Specify the SQL mode (section 4). Default is process mode.
`-c`	Specify the Oracle Database service mode. Default is connect string.
`-f`	Specify the detailed format (132 columns). Default is standard (80 columns).
`-b`	Specify the batch mode. Default is text-based user interface.
`-n`	Specify the maximum number of iterations.
`-i`	Specify the interval delay in seconds. Default is 5 seconds.

Examples

The following command examples demonstrate how to use Oracle Trace File Analyzer to search collected data:

-
  ```
  $ tfactl analyze -search "error" -last 2d
  ```

  Oracle Trace File Analyzer searches alert and system log files from the past two days for messages that contain the case-insensitive string "error".

-
  ```
  $ tfactl analyze -comp os -for "Jul/01/2016 11" -search "."
  ```

  Oracle Trace File Analyzer displays all system log messages for July 1, 2016 at 11 am.

-
  ```
  $ tfactl analyze -search "/ORA-/c" -comp db -last 2d
  ```

  Oracle Trace File Analyzer searches database alert logs for the case-sensitive string "ORA-" from the past two days.

The following command examples demonstrate how to use Oracle Trace File Analyzer to analyze collected data:

-
  ```
  $ tfactl analyze -last 5h
  ```

Oracle Trace File Analyzer displays a summary of events collected from all alert logs and system messages from the past five hours.

- `$ tfactl analyze -comp os -last 1d`

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

- `$ tfactl analyze -last 1h -type generic`

  Oracle Trace File Analyzer analyzes all generic messages from the last hour.

The following command examples demonstrate how to use Oracle Trace File Analyzer to analyze `OSWatcher` Top and Slabinfo:

- `$ tfactl analyze -comp osw -last 6h`

  Oracle Trace File Analyzer displays `OSWatcher` Top summary for the past six hours.

- `$ tfactl analyze -comp oswslabinfo -from "2016-07-01" -to "2016-07-03"`

  Oracle Trace File Analyzer displays `OSWatcher` Slabinfo summary for specified time period.

E.3.2 tfactl changes

Use the `tfactl changes` command to view the changes detected by Oracle Trace File Analyzer.

Syntax

```
tfactl changes
[-from time -to time | -for time | last time_length]
```

Parameters

Option	Description

`from` *`time`* `-to` *`time`*	Specify the `-from` and `-to` parameters (you must use these two parameters together) to view changes that occurred during a specific time interval.
	Supported time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`

`for` *`time`*	Specify the `-for` parameter to view the changes that occurred at the time given.
	Supported time formats:
	`"Mon/dd/yyyy"`
	`"yyyy-mm-dd"`

`-last` *`nh|d`*	Specify the `-last` parameter to view changes for the past specific number of hours (h), or days (d).

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%
```

E.3.3 tfactl events

Use the `tfactl events` command to view the events detected by Oracle Trace File Analyzer.

Syntax

```
tfactl events
[-search keyword | -component ASM|CRS | -database db_name | -instance
db_instance_name | -source filename | -from time -to time | -json | -fields
all|fields_list]
```

Parameters

Option	Description
`component [ASM|CRS]`	Searches all Oracle Automatic Storage Management (Oracle ASM) or Oracle Clusterware events.
`database` *`db_name`*	Specify the name of an Oracle Database to search all events from that Oracle Database.
`instance` *`db_instance_name`*	Specify the name of an Oracle Database instance to search all events from that Oracle Database instance.
`source` *`filename`*	Specify the source file name to search all events from that alert file.
`json`	Displays event information in JSON format.

`-last` *`nh|d`* `| -from` *`time`* `-to` *`time`* `| -for` *`time`*`]`	• Specify the `-last` parameter to view events for the past specific number of hours (*h*) or days (*d*).
	• Specify the `-from` and `-to` parameters (you must use these two parameters together) to view events that occurred during a specific time interval.
	Supported time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`
	• Specify the `-for` parameter to view events for the time given.
	Supported time formats:
	`"Mon/dd/yyyy"`
	`"yyyy-mm-dd"`

> **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.

Option	Description
`fields all|`*`fields_list`*	When provided with the `-json` option, the command returns only the requested fields

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
```

E.3.4 tfactl isa

Use the `tfactl isa` command to view the Infrastructure Service Automation (ISA) score.

Syntax

```
tfactl isa
[-availability]
[-all]
[-node all|local|n1,n2,...]
```

Parameters

Table E-31 tfactl run Command Parameters

Parameter	Description
`availability`	Includes the Availability Score.
`all`	Displays all the details.
`node`	Specify a comma-separated list of host names.

E.3.5 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 E-32 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 E-33 tfactl run Analysis Tools Parameters

Parameter	Description
`orachk`	Runs `Oracle ORAchk`.
`oratop`	Runs `oratop`.
`oswbb`	Runs `OSWatcher Analyzer`.
`prw`	Runs `Procwatcher`.
`alertsummary`	Prints summary of important events in Oracle Database / ASM alert logs.
`calog`	Prints Oracle Clusterware activity logs.
`dbglevel`	Sets Oracle Clusterware log / trace levels using profiles.
`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.
`managelogs`	Purge slogs.
`menu`	Oracle Trace File Analyzer Collector menu system.
`param`	Prints parameter value.
`ps`	Finds a process.
`pstack`	Runs `pstack` on a process.
`summary`	Prints system summary.
`tail`	Tails log files.
`triage`	Summarize `OSWatcher` / `ExaWatcher` data.
`vi`	Searches and opens files in the `vi` editor.

Profiling Tools

Table E-34 tfactl run Profiling Tools Parameters

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

E.3.6 tfactl search

Use the `tfactl search` command to search all metadata stored in the Oracle Trace File Analyzer index.

Syntax

```
tfactl search
[-json json_string | -fields all|fields_list | -showdatatypes | -showfields
datatype]
```

Parameters

Table E-35 tfactl search Command Parameters

Parameter	Description
`json`	JSON string containing the search criteria.
`fields`	Returns the JSON output with only the requested fields.
`showdatatypes`	Displays the list of all available datatypes.
`showfields`	Displays the list of fields available in a datatype.

E.3.7 tfactl summary

Use the `tfactl summary` command to view the summary of Oracle Trace File Analyzer deployment.

Syntax

```
tfactl [run] summary [OPTIONS]
```

Options

Option	Description
`[no_components]`	[Default] Complete summary collection
`-overview`	[Optional/Default] Complete summary collection - overview.
`-crs`	[Optional/Default] Oracle Clusterware status summary.
`-asm`	[Optional/Default] Oracle ASM status summary.
`-acfs`	[Optional/Default] Oracle ACFS status Summary.
`-database`	[Optional/Default] Oracle Database status summary.
`-exadata`	[Optional/Default] Oracle Exadata status summary. Not enabled/ignored in Microsoft Windows and Non-Exadata machine
`-patch`	[Optional/Default] Patch details.
`-listener`	[Optional/Default] LISTENER status summary.
`-network`	[Optional/Default] NETWORK status summary.
`-os`	[Optional/Default] Operating system status summary.

Option	Description
`-tfa`	[Optional/Default] Oracle Trace File Analyzer status summary.
`-summary`	[Optional/Default] Summary tool metadata.
`-json`	[Optional] - Prepare JSON report.
`-html`	[Optional] - Prepare HTML report.
`-print`	[Optional] - Display [HTML or JSON] report at console.
`-silent`	[Optional] - Interactive console by default.
`-history` *`num`*	[Optional] - View Previous *numberof* summary collection history in interpreter.
`-node`	*node(s)* : [Optional] - local or comma-separated list of names of nodes.
`-help`	Usage/help

E.3.8 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 E-25 tfactl toolstatus

The `tfactl toolstatus` command returns output similar to the following, showing which tool is deployed and where the tool is deployed.

```
.------------------------------------------------------------------.
|                 TOOLS STATUS - HOST : myhost                     |
+----------------------+--------------+--------------+-------------+
| Tool Type            | Tool         | Version      | Status      |
+----------------------+--------------+--------------+-------------+
| Development Tools    | orachk       |   12.2.0.1.3 | DEPLOYED    |
|                      | oratop       |       14.1.2 | DEPLOYED    |
+----------------------+--------------+--------------+-------------+
| Support Tools Bundle | darda        | 2.10.0.R6036 | DEPLOYED    |
|                      | oswbb        |        8.1.2 | RUNNING     |
|                      | prw          | 12.1.13.11.4 | NOT RUNNING |
+----------------------+--------------+--------------+-------------+
| TFA Utilities        | alertsummary |   12.2.1.1.0 | DEPLOYED    |
|                      | calog        |   12.2.0.1.0 | DEPLOYED    |
|                      | dbcheck      |   18.3.0.0.0 | DEPLOYED    |
|                      | dbglevel     |   12.2.1.1.0 | DEPLOYED    |
|                      | grep         |   12.2.1.1.0 | DEPLOYED    |
|                      | history      |   12.2.1.1.0 | DEPLOYED    |
|                      | ls           |   12.2.1.1.0 | DEPLOYED    |
|                      | managelogs   |   12.2.1.1.0 | DEPLOYED    |
|                      | menu         |   12.2.1.1.0 | DEPLOYED    |
|                      | param        |   12.2.1.1.0 | DEPLOYED    |
|                      | ps           |   12.2.1.1.0 | DEPLOYED    |
|                      | pstack        |   12.2.1.1.0 | DEPLOYED    |
|                      | summary       |   12.2.1.1.0 | DEPLOYED    |
|                      | tail          |   12.2.1.1.0 | DEPLOYED    |
|                      | triage        |   12.2.1.1.0 | DEPLOYED    |
|                      | vi            |   12.2.1.1.0 | DEPLOYED    |
'----------------------+--------------+--------------+-------------'

Note :-
  DEPLOYED    : Installed and Available - To be configured or run 
interactively.
  NOT RUNNING : Configured and Available - Currently turned off interactively.
  RUNNING     : Configured and Available.
```

E.4 Running Oracle Trace File Analyzer Diagnostic Collection Commands

Run the diagnostic collection commands to collect diagnostic data.

- tfactl collection
  Use the `tfactl collection` command to stop a running Oracle Trace File Analyzer collection.
- tfactl dbglevel
  Use the `tfactl dbglevel` command to set Oracle Grid Infrastructure trace levels.
- tfactl diagcollect
  Use the `tfactl diagcollect` command to perform on-demand diagnostic collection.
- tfactl diagcollect -srdc
  Use the `tfactl diagcollect -srdc` command to run a Service Request Data Collection (SRDC).
- tfactl directory
  Use the `tfactl directory` command to add a directory to, or remove a directory from the list of directories to analyze their trace or log files.
- tfactl ips
  Use the `tfactl ips` command to collect Automatic Diagnostic Repository diagnostic data.
- tfactl managelogs
  Use the `tfactl managelogs` command to manage Automatic Diagnostic Repository log and trace files.
- 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.

E.4.1 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.

E.4.2 tfactl dbglevel

Use the `tfactl dbglevel` command to set Oracle Grid Infrastructure trace levels.

Syntax

```
tfactl [run] dbglevel
[ {-set|-unset} profile_name
-dependency [dep1,dep2,...|all]
-dependency_type [type1,type2,type3,...|all]
| {-view|-drop} profile_name
| -lsprofiles
| -lsmodules
| -lscomponents [module_name]
| -lsres
| -create profile_name [ -desc description
| [-includeunset] [-includetrace]
| -debugstate | -timeout time ]
| -modify profile_name [-includeunset] [-includetrace]
| -getstate [ -module module_name ]
| -active [profile_name]
| -describe [profile_name] ] ]
```

Parameters

Table E-36 tfactl dbglevel Command Parameters

Parameter	Description
`profile_name`	Specify the name of the profile.
`active`	Displays the list of active profiles.
`set`	Sets the trace or log levels for the profile specified.
`unset`	Unsets the trace or log levels for the profile specified.
`view`	Displays the trace or log entries for the profile specified.
`create`	Creates a profile.
`drop`	Drops the profile specified.
`modify`	Modifies the profile specified.
`describe`	Describes the profiles specified.
`lsprofiles`	Lists all the available profiles.
`lsmodules`	Lists all the discovered Oracle Clusterware modules.
`lscomponents`	Lists all the components associated with the Oracle Clusterware module.
`lsres`	Lists all the discovered Oracle Clusterware resources.
`getstate`	Displays the current trace or log levels for the Oracle Clusterware components or resources.
`module`	Specify the Oracle Clusterware module.

Table E-36 (Cont.) tfactl dbglevel Command Parameters

Parameter	Description
`dependency`	Specify the dependencies to consider, start, or stop dependencies, or both.
`dependency_type`	Specify the type of dependencies to be consider.
`debugstate`	Generates a System State Dump for all the available levels.
`includeunset`	Adds or modifies an unset value for the Oracle Clusterware components or resources.
`includetrace`	Adds or modifies a trace value for the Oracle Clusterware components.

> **WARNING:**
>
> Set the profiles only at the direction of Oracle Support.

E.4.3 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

Syntax

```
tfactl diagcollect [[component_name1] [component_name2] ... [component_nameN]
| [-srdc srdc_profile] | [-defips]]
[-sr SR#]
[-node all|local|n1,n2,...]
[-tag tagname]
[-z filename]
[-last nh|d | -from time -to time | -for time]
[-nocopy]
[-notrim]
[-silent]
[-cores]
[-collectalldirs]
[-collectdir dir1,dir2...]
[-examples]
[-upload]
[-id]

Components:-ips|-database|-asm|-crsclient|-dbclient|-dbwlm|-tns|-rhp|-
procinfo|-afd|-crs|-cha|-wls|-emagent|-oms|-ocm|-emplugins|-em|-acfs
|-install|-cfgtools|-os|-ashhtml|-ashtext|-awrhtml|-awrtext
```

Parameters

Prefix each option with a minus sign (-).

Option	Description
`[[`*component_name1*`] [`*component_name2*`] ... [`*component_nameN*`] | [-srdc` *srdc_profile*`] | [-defips]]]`	Specify the list of components for which you want to obtain collections, or specify the SRDC name, or specify to include Incident Packaging Service (IPS) Packages for Oracle Automatic Storage Management (Oracle ASM), Oracle Clusterware, and Oracle Databases in the default collection.
`[-sr` *SR#*`]`	Specify the Service Request number to which Oracle Trace File Analyzer automatically uploads all 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.

`[-last` *nh|d* `| -from` *time* `-to` *time* `| -for` *time*`]`	• Specify the `-last` 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.
	You can also use `-since`, which has the same functionality as `-last`. This option is included for backward compatibility.
	• 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.
	Supported time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`
	• 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.
	Supported time formats:
	`"Mon/dd/yyyy"`
	`"yyyy-mm-dd"`

> **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.

Option	Description
`-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
`-cores`	Specify this parameter to collect core files when it would normally have not 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.
`-upload` `-id`	`tfactl diagcollect` *`args`* `-upload` *`config_name`* `-id` *`identifier`* Generates diagnostic collection and uploads to the specified end point name or config name to the location identifier. Optionally, you can specify the identifier. If you are uploading to My Oracle Support, then specify the SR number. `tfactl diagcollect -since 1h -upload` *`config_name`* `-id` *`identifier`* For example: `$ tfactl diagcollect -since 1h -upload mos -id` *`3-123456789`*

Related Topics

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

E.4.4 tfactl diagcollect -srdc

Use the `tfactl diagcollect -srdc` command to run a Service Request Data Collection (SRDC).

Syntax

```
tfactl diagcollect -srdc srdc_profile
[-tag tagname]
[-z filename]
[-last nh|d | -from time -to time | -for date]
-database database
```

Parameters

Each option must be prefixed with a minus sign (-).

Option	Description
`[-srdc` *`srdc_profile`*`]`	Specify the SRDC profile.

Option	Description
`-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.

`[-last` *`nh`*`|`*`d`* `| -from` *`time`* `-to` *`time`* `| -for` *`date`*`]`	• Specify the `-last` 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.
	You can also use `-since`, which has the same functionality as `-last`. This option is included for backward compatibility.
	• 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.
	Supported time formats:
	`"Mon/dd/yyyy hh:mm:ss"`
	`"yyyy-mm-dd hh:mm:ss"`
	`"yyyy-mm-ddThh:mm:ss"`
	`"yyyy-mm-dd"`
	• Specify the `-for` parameter to collect files that have relevant data for the date specified. 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.
	Supported time formats:
	`"Mon/dd/yyyy"`
	`"yyyy-mm-dd"`
	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.

`-database` *`database`*	Specify the name of the database.

SRDC Profiles

SRDC Profile	Description
`listener_services`	Collects data for listener services errors: `TNS-12514` / `TNS-12516` / `TNS-12518` / `TNS-12519` / `TNS-12520` / `TNS-12528`.
`naming_services`	Collects data for naming services errors: `ORA-12514` / `ORA-12528`.
`ORA-00020`	Collects data regarding maximum number of processes exceeded.
`ORA-00060`, `ORA-00600`	Collects data for internal errors.
`ORA-00700`	Collects data for soft internal error.

SRDC Profile	Description
ORA-01031	Collects standard information for ORA-1031 / ORA-1017 during SYSDBA connections
ORA-01555	Collects data for Oracle Database Snapshot too old error.
ORA-01578	Collects data for NOLOGGING ORA-1578 / ORA-26040 DBV-00201.
ORA-01628	Collects data for Oracle Database Snapshot too old error.
ORA-04030	Collects data for OS process private memory was exhausted error.
ORA-04031	Collects data for More shared memory is needed in the shared/streams pool. error.
ORA-07445	Collects data for Exception encountered, core dump. error.
ORA-08102	Collects data for ORA error ORA-08102.
ORA-08103	Collects data for ORA error ORA-08103.
ORA-27300	Collects data for OS system dependent operation: open failed with status: (status). error.
ORA-27301	Collects data for OS failure message: (message). error.
ORA-27302	Collects data for Failure occurred at: (module). error.
ORA-30036	Collects data for Oracle Database Unable to extend Undo Tablespace error.
dbasm	Collects data for Oracle Database storage problems.
dbaudit	Collects standard information for Oracle Database auditing.
dbawrspace	Collects data for Oracle Database Automatic Workload Repository (AWR) space problems.
dbexp	Collects information for troubleshooting original Export (exp) related problems.
dbexpdp	Collects data for Data Pump Export generic issues.
dbexpdpapi	Collects data for Data Pump Export API Issues.
dbexpdpperf	Collects data for Data Pump Export performance issues.
dbexpdptts	Collects data to supply for Transportable Tablespace Data Pump and original EXPORT, IMPORT.
dbfs	Collects data for dbfs issues.
dbggclassicmode	Collects data for Oracle GoldenGate Classic Mode issues.
dbggintegratedmode	Collects data for Oracle GoldenGate Extract / Replicat abends problems.
dbimp	Collects data for troubleshooting original Import (imp) releated problems.
dbimpdp	Collects data for Data Pump Import generic issues.
dbimpdpperf	Collects data for Data Pump Import performance issues.
dbinstall	Collects data for Oracle Database install / upgrade problems.
dbpartition	Collects data for Create / maintain partitioned / subpartitioned table / index problems.
dbpartitionperf	Collects data for slow Create / Alter / Drop commands against partitioned table / index problems.
dbpatchconflict	Collects data for Oracle Database patch conflict problems.
dbpatchinstall	Collects data for Oracle Database patch install problems.

SRDC Profile	Description
dbperf	Collects data for Oracle Database performance problems.
dbpreupgrade	Collects data for Oracle Database preupgrade problems.
dbrman	Collects data for RMAN related issues, such as backup, maintenance, restore and recover, `RMAN-08137`, or `RMAN-08120`.
dbrman600	Collects data for `RMAN-00600` error (My Oracle Support note 2045195.1).
dbrmanperf	Collects data for RMAN Performance error (My Oracle Support note 1671509.1).
dbscn	Collects data for Oracle Database SCN problems.
dbshutdown	Collects data for single instance Oracle Database shutdown problems.
dbsqlperf	Collects data for an SQL performance problem using Oracle Trace File Analyzer Collector.
dbstartup	Collects data for single instance Oracle Database startup problems.
dbtde	Collects data for Transparent Data Encryption (TDE) (My Oracle Support note 1905607.1)
dbundocorruption	Collects data for UNDO corruption problems.
dbunixresources	Collects data for Oracle Database issues related to operating system resources.
dbupgrade	Collects data for Oracle Database upgrade problems.
dbxdb	Collects data Oracle Database XDB installation and invalid object problems.
dnfs	Collects data for DNFS problems.
emagentperf	Collects data for Enterprise Manager Agent performance issues.
emcliadd	Collects data for Enterprise Manager errors while adding an Oracle Database, a listener, or an ASM target using Enterprise Manager command-line.
emclusdisc	Collects data for cluster target, cluster (RAC) Oracle Database, or an ASM target is not discovered issue.
emdbsys	Collects data for Enterprise Manager Oracle Database system target is not discovered, detected, removed, or renamed correctly issue.
emdebugoff	Collects data for unsetting Enterprise Manager debug.
emdebugon	Collects data for setting Enterprise Manager debug.
emgendisc	Collects data for Enterprise Manager generic error while discovering, or removing an Oracle Database, a listener, or an ASM target.
emmetricalert	Collects data for Enterprise Manager metric events not raised and general metric alert related issues.
emomscrash	Collects for all Enterprise Manager OMS crash or restart performance issues.
emomsheap	Collects data for Enterprise Manager OMS heap usage alert performance issues.
emomshungcpu	Collects data for Enterprise Manager OMS hung or high CPU usage performance issues.
emprocdisc	Collects data for Enterprise Manager Oracle Database, listener, or an ASM target is not discovered or detected by the discovery process issues.
emrestartoms	Collects data for Enterprise Manager restart OMS crash problems.

SRDC Profile	Description
`emtbsmetric`	Collects data for Enterprise Manager Tablespace space used metric issues.
`esexalogic`	Collects data for Oracle Exalogic Full Exalogs problems.
`ggintegratedmodenodb`	Collects data for Oracle GoldenGate Extract/Replicat abends problems.
`internalerror`	Collects data for all other types of internal Oracle Database errors.

Related Topics

- https://support.oracle.com/rs?type=doc&amp;id=2175568.1
- https://support.oracle.com/rs?type=doc&amp;id=2045195.1
- https://support.oracle.com/rs?type=doc&amp;id=1671509.1
- https://support.oracle.com/rs?type=doc&amp;id=1905607.1

E.4.5 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 E-37 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 DB (BDB) 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 Grid Infrastructure
- 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.

Example E-26 tfactl directory

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
```

E.4.6 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]
```

For detailed help on each topic use:

```
help ips topic
```

Parameters

Table E-38 tfactl ips Command Parameters

Parameter	Description
`ADD`	Adds incidents to an existing package.
`ADD FILE`	Adds a file to an existing package.
`ADD NEW INCIDENTS`	Finds new incidents for the problems and add the latest ones to the package.
`CHECK REMOTE KEYS`	Creates a file with keys matching incidents in specified package.
`COPY IN FILE`	Copies an external file into Automatic Diagnostic Repository, and associates it with a package and (optionally) an incident.
`COPY OUT FILE`	Copies an Automatic Diagnostic Repository file to a location outside Automatic Diagnostic Repository.
`CREATE PACKAGE`	Creates a package, and optionally select contents for the package.
`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.

Table E-38 (Cont.) tfactl ips Command Parameters

Parameter	Description
`SHOW CONFIGURATION`	Shows the current Incident Packaging Service settings.
`SHOW FILES`	Shows the files included in the specified package.
`SHOW INCIDENTS`	Shows incidents included in the specified package.
`SHOW PROBLEMS`	Shows problems for the current Automatic Diagnostic Repository home.
`SHOW PACKAGE`	Shows details for the specified package.
`UNPACK FILE`	Unpackages a physical file into the specified path.
`UNPACK PACKAGE`	Unpackages physical files in the current directory into the specified path, if they match the package name.
`USE REMOTE KEYS`	Adds incidents matching the keys in the specified file to the specified package.

- tfactl ips ADD
  Use the `tfactl ips ADD` command to add incidents to an existing package.
- tfactl ips ADD FILE
  Use the `tfactl ADD FILE` command to add a file to an existing package.
- tfactl ips ADD NEW INCIDENTS
  Use the `tfactl ips ADD NEW INCIDENTS` command to find new incidents for the problems in a specific package, and add the latest ones to the package.
- tfactl ips CHECK REMOTE KEYS
  Use the `tfactl ips CHECK REMOTE KEYS` command to create a file with keys matching incidents in a specified package.
- tfactl ips COPY IN FILE
  Use the `tfactl ips COPY IN FILE` command to copy an external file into Automatic Diagnostic Repository, and associate the file with a package and (optionally) an incident.
- tfactl ips COPY OUT FILE
  Use the `tfactl ips COPY OUT FILE` command to copy an Automatic Diagnostic Repository file to a location outside Automatic Diagnostic Repository.
- tfactl ips CREATE PACKAGE
  Use the `tfactl ips CREATE PACKAGE` command to create a package, and optionally select the contents for the package.
- tfactl ips DELETE PACKAGE
  Use the `tfactl ips DELETE PACKAGE` command to drop a package and its contents from the Automatic Diagnostic Repository.
- tfactl ips FINALIZE PACKAGE
  Use the `tfactl ips FINALIZE PACKAGE` command to get a package ready for shipping by automatically including correlated contents.
- tfactl ips GENERATE PACKAGE
  Use the `tfactl ips GENERATE PACKAGE` command to create a physical package (`zip` file) in the target directory.
- tfactl ips GET MANIFEST
  Use the `tfactl ips GET MANIFEST` command to extract the manifest from a package file and view it.

- tfactl ips GET METADATA
  Use the `tfactl ips GET METADATA` command to extract the metadata XML document from a package file and view it.
- tfactl ips GET REMOTE KEYS
  Use the `tfactl ips GET REMOTE KEYS` command to create a file with keys matching incidents in a specific package.
- tfactl ips PACK
  Use the `tfactl ips PACK` command to create a package and immediately generate the physical package.
- tfactl ips REMOVE
  Use the `tfactl ips REMOVE` command to remove incidents from an existing package.
- tfactl ips REMOVE FILE
  Use the `tfactl ips REMOVE FILE` command to remove a file from an existing package.
- tfactl ips SET CONFIGURATION
  Use the `tfactl ips SET CONFIGURATION` command to change the value of an Incident Packaging Service configuration parameter.
- tfactl ips SHOW CONFIGURATION
  Use the `tfactl ips SHOW CONFIGURATION` command to view the current Incident Packaging Service settings.
- tfactl ips SHOW FILES
  Use the `tfactl ips SHOW FILES` command to view the files included in a specific package.
- tfactl ips SHOW INCIDENTS
  Use the `tfactl ips SHOW INCIDENTS` command to view the incidents included in a specific package.
- tfactl ips SHOW PROBLEMS
  Use the `tfactl ips SHOW PROBLEMS` command to view the problems for the current Automatic Diagnostic Repository home.
- tfactl ips SHOW PACKAGE
  Use the `tfactl ips SHOW PACKAGE` command to view the details of a specific package.
- tfactl ips UNPACK FILE
  Use the `tfactl ips UNPACK FILE` command to unpack a physical file into a specific path.
- tfactl ips UNPACK PACKAGE
  Use the `tfactl ips UNPACK PACKAGE` command to unpack physical files in the current directory into a specific path, if they match the package name.
- tfactl ips USE REMOTE KEYS
  Use the `tfactl ips USE REMOTE KEYS` command to add incidents matching the keys in a specific file to a specific package.

E.4.6.1 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 E-39 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.

Example E-27 tfactl ips ADD

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

E.4.6.2 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 E-40 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.

Example E-28 tfactl ips ADD FILE

```
$ tfactl ips add file ADR_HOME/trace/mydb1_ora_13579.trc package 12
```

E.4.6.3 tfactl ips ADD NEW INCIDENTS

Use the `tfactl ips ADD NEW INCIDENTS` 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_id
```

Parameters

Table E-41 tfactl ips ADD NEW INCIDENTS Command Parameters

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

E.4.6.4 tfactl ips CHECK REMOTE KEYS

Use the `tfactl ips CHECK REMOTE KEYS` command to create a file with keys matching incidents in a specified package.

Syntax

```
tfactl ips CHECK REMOTE KEYS file_spec PACKAGE package_id
```

Parameters

Table E-42 tfactl ips CHECK REMOTE KEYS Command Parameters

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

E.4.6.5 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 E-43 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.

Example E-29 tfactl ips COPY IN FILE

```
$ tfactl ips copy in file /tmp/key_file.txt to new_file.txt package 12
incident 62
```

E.4.6.6 tfactl ips COPY OUT FILE

Use the `tfactl ips COPY OUT FILE` command to copy an Automatic Diagnostic Repository file to a location outside Automatic Diagnostic Repository.

Syntax

```
tfactl IPS COPY OUT FILE source TO target [OVERWRITE]
```

Parameters

Table E-44 tfactl ips COPY OUT FILE Command Parameters

Parameter	Description
source	Specify the file with file name and full path (full or relative). This file must be inside ADR.
target	Specify the file with file name and full path (full or relative). This file must be outside ADR.

Options

`OVERWRITE`: If the file exists, then use the `OVERWRITE` option to overwrite the file.

Example E-30 tfactl ips COPY OUT FILE

```
$ tfactl ips copy out file ADR_HOME/trace/ora_26201 to /tmp/trace_26201.txt
```

E.4.6.7 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 E-45 tfactl ips CREATE PACKAGE Command Parameters

Parameter	Description
incid	Specify the ID of the incident to use for selecting the package contents.

Table E-45   (Cont.) tfactl ips CREATE PACKAGE Command Parameters

Parameter	Description
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 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 E-31   tfactl ips CREATE PACKAGE

```
$ 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'
```

E.4.6.8 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 E-46 tfactl ips DELETE PACKAGE Command Parameters

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

Example E-32 tfactl ips DELETE PACKAGE

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

E.4.6.9 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
```

Example E-33 tfactl ips FINALIZE PACKAGE

```
$ tfactl ips finalize package 12
```

E.4.6.10 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 E-47 tfactl ips GENERATE PACKAGE Command Parameters

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

Table E-47 (Cont.) tfactl ips GENERATE PACKAGE Command Parameters

Parameter	Description
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 E-34 tfactl ips GENERATE PACKAGE

```
$ tfactl ips generate package 12 in /tmp
```

E.4.6.11 tfactl ips GET MANIFEST

Use the `tfactl ips GET MANIFEST` command to extract the manifest from a package file and view it.

Syntax

```
tfactl ips GET MANIFEST FROM FILE file
```

Parameters

Table E-48 tfactl ips GET MANIFEST FROM FILE Command Parameters

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

Example E-35 tfactl ips GET MANIFEST

```
$ tfactl ips get manifest from file /tmp/IPSPKG_200704130121_COM_1.zip
```

E.4.6.12 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]
```

Parameters

Table E-49 tfactl ips GET METADATA Command Parameters

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

Example E-36 tfactl ips GET METADATA

```
$ tfactl ips get metadata from file /tmp/IPSPKG_200704130121_COM_1.zip
```

E.4.6.13 tfactl ips GET REMOTE KEYS

Use the `tfactl ips GET REMOTE KEYS` 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 E-50 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 E-37 tfactl ips GET REMOTE KEYS

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

E.4.6.14 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 E-51 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 E-38 tfactl ips PACK

```
$ 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'
```

E.4.6.15 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 E-52 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 E-39 tfactl ips REMOVE

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

E.4.6.16 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
```

Parameters

Table E-53 tfactl ips REMOVE 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 remove the file from.

Example E-40 tfactl ips REMOVE FILE

```
$ tfactl ips remove file ADR_HOME/trace/mydb1_ora_13579.trc package 12
```

E.4.6.17 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 E-54 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 E-41 tfactl ips SET CONFIGURATION

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

E.4.6.18 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
```

Example E-42 tfactl ips SHOW CONFIGURATION

```
$ tfactl ips show configuration

Multiple ORACLE HOMES were found, please select one ...

option[0] /scratch/app/oradb/product/11.2.0/dbhome_11204
option[1] /scratch/app/11.2.0.4/grid

Pls select an ORACLE_HOME to be used for the ADRCI binary [0] ?0
/scratch/app/oradb/product/11.2.0/dbhome_11204 was selected


Multiple ADR basepaths were found, please select one ...

( ) option[0] /scratch/app/oradb
( ) option[1] /scratch/app/oragrid
```

```
Pls select an ADR basepath [0..1] ?0
/scratch/app/oradb was selected


Multiple ADR homepaths were found for /scratch/app/oradb, please select
one ...

( ) option[0] diag/rdbms/racone/racone_2
( ) option[1] diag/rdbms/rdb11204/rdb112041
( ) option[2] diag/rdbms/ogg11204/ogg112041
( ) option[3] diag/rdbms/apxcmupg/apxcmupg_1
( ) option[4] diag/rdbms/apxcmupg/apxcmupg_2
    option[5] Done

Pls select a homepath [5] ?0
diag/rdbms/racone/racone_2 was selected

PARAMETER INFORMATION:
   PARAMETER_ID           1
   NAME                   CUTOFF_TIME
   DESCRIPTION            Maximum age for an incident to be considered for
inclusion
   UNIT                   Days
   VALUE                  90
   DEFAULT_VALUE          90
   MINIMUM                1
   MAXIMUM                4294967295
   FLAGS                  0

PARAMETER INFORMATION:
   PARAMETER_ID           2
   NAME                   NUM_EARLY_INCIDENTS
   DESCRIPTION            How many incidents to get in the early part of the
range
   UNIT                   Number
   VALUE                  3
   DEFAULT_VALUE          3
   MINIMUM                1
   MAXIMUM                4294967295
   FLAGS                  0

PARAMETER INFORMATION:
   PARAMETER_ID           3
   NAME                   NUM_LATE_INCIDENTS
   DESCRIPTION            How many incidents to get in the late part of the
range
   UNIT                   Number
   VALUE                  3
   DEFAULT_VALUE          3
   MINIMUM                1
   MAXIMUM                4294967295
   FLAGS                  0

PARAMETER INFORMATION:
   PARAMETER_ID           4
   NAME                   INCIDENT_TIME_WINDOW
   DESCRIPTION            Incidents this close to each other are considered
correlated
   UNIT                   Minutes
   VALUE                  5
   DEFAULT_VALUE          5
   MINIMUM                1
   MAXIMUM                4294967295
   FLAGS                  0

PARAMETER INFORMATION:
   PARAMETER_ID           5
   NAME                   PACKAGE_TIME_WINDOW
   DESCRIPTION            Time window for content inclusion is from x hours
before first included incident to x hours after last incident
   UNIT                   Hours
   VALUE                  24
   DEFAULT_VALUE          24
   MINIMUM                1
   MAXIMUM                4294967295
   FLAGS                  0

PARAMETER INFORMATION:
   PARAMETER_ID           6
   NAME                   DEFAULT_CORRELATION_LEVEL
   DESCRIPTION            Default correlation level for packages
   UNIT                   Number
   VALUE                  2
   DEFAULT_VALUE          2
   MINIMUM                1
   MAXIMUM                4
   FLAGS                  0
```

E.4.6.19 tfactl ips SHOW FILES

Use the `tfactl ips SHOW FILES` command to view the files included in a specific package.

Syntax

```
tfactl ips SHOW FILES PACKAGE package_id
```

Example E-43 tfactl ips SHOW FILES

```
$ tfactl ips show files package 12
```

E.4.6.20 tfactl ips SHOW INCIDENTS

Use the `tfactl ips SHOW INCIDENTS` command to view the incidents included in a specific package.

Syntax

```
tfactl ips SHOW INCIDENTS PACKAGE package_id
```

Example E-44 tfactl ips SHOW INCIDENTS

```
$ tfactl ips show incidents package 12
```

E.4.6.21 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
```

Example E-45 tfactl ips SHOW PROBLEMS

```
tfactl ips show problems

Multiple ADR basepaths were found, please select one ...

( ) option[0] /scratch/app/oradb
( ) option[1] /scratch/app/oragrid

Pls select an ADR basepath [0..1] ?0
/scratch/app/oradb was selected


ADR Home = /scratch/app/oradb/diag/rdbms/racone/racone_2:
*************************************************************************
0 rows fetched

ADR Home = /scratch/app/oradb/diag/rdbms/rdb11204/rdb112041:
*************************************************************************
PROBLEM_ID
PROBLEM_KEY
LAST_INCIDENT        LASTINC_TIME
--------------------
-----------------------------------------------------------
-------------------- ----------------------------------------
2                    ORA 700
[kgerev1]                                           42605
2016-07-05 07:53:28.578000 -07:00
1                    ORA
600                                                     42606
2016-07-05 07:53:30.427000 -07:00

ADR Home = /scratch/app/oradb/diag/rdbms/ogg11204/ogg112041:
*************************************************************************
PROBLEM_ID
PROBLEM_KEY
LAST_INCIDENT        LASTINC_TIME
--------------------
-----------------------------------------------------------
-------------------- ----------------------------------------
3                    ORA
```

```
4030                                                    51504
2017-09-26 10:03:03.922000 -07:00
2                    ORA 700
[kgerev1]                                           54401
2017-09-26 10:03:10.371000 -07:00
1                    ORA
600                                                     54402
2017-09-26 10:03:11.446000 -07:00
6                    ORA 600
[gc_test_error]                                     54691
2017-10-23 03:03:40.599000 -07:00
5                    ORA
4031                                                    64277
2017-12-13 04:48:16.035000 -08:00
4                    ORA
7445                                                    96286
2018-05-29 08:26:11.326000 -07:00

ADR Home = /scratch/app/oradb/diag/rdbms/apxcmupg/apxcmupg_1:
*************************************************************************
0 rows fetched

ADR Home = /scratch/app/oradb/diag/rdbms/apxcmupg/apxcmupg_2:
*************************************************************************
0 rows fetched
```

E.4.6.22 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 E-46 tfactl ips SHOW PACKAGE

```
$ tfactl ips show package

Multiple ADR basepaths were found, please select one ...
```

```
( ) option[0] /scratch/app/oradb
( ) option[1] /scratch/app/oragrid

Pls select an ADR basepath [0..1] ?0
/scratch/app/oradb was selected


Multiple ADR homepaths were found for /scratch/app/oradb, please select
one ...

( ) option[0] diag/rdbms/racone/racone_2
( ) option[1] diag/rdbms/rdb11204/rdb112041
( ) option[2] diag/rdbms/ogg11204/ogg112041
( ) option[3] diag/rdbms/apxcmupg/apxcmupg_1
( ) option[4] diag/rdbms/apxcmupg/apxcmupg_2
    option[5] Done

Pls select a homepath [5] ?1
diag/rdbms/rdb11204/rdb112041 was selected

   PACKAGE_ID             1
   PACKAGE_NAME           IPSPKG_20160731165615
   PACKAGE_DESCRIPTION
   DRIVING_PROBLEM        N/A
   DRIVING_PROBLEM_KEY    N/A
   DRIVING_INCIDENT       N/A
   DRIVING_INCIDENT_TIME  N/A
   STATUS                 Generated (4)
   CORRELATION_LEVEL      Typical (2)
   PROBLEMS               0 main problems, 0 correlated problems
   INCIDENTS              0 main incidents, 0 correlated incidents
   INCLUDED_FILES         27

   PACKAGE_ID             2
   PACKAGE_NAME           IPSPKG_20160731170111
   PACKAGE_DESCRIPTION
   DRIVING_PROBLEM        N/A
   DRIVING_PROBLEM_KEY    N/A
   DRIVING_INCIDENT       N/A
   DRIVING_INCIDENT_TIME  N/A
   STATUS                 Generated (4)
   CORRELATION_LEVEL      Typical (2)
   PROBLEMS               0 main problems, 0 correlated problems
   INCIDENTS              0 main incidents, 0 correlated incidents
   INCLUDED_FILES         27

   PACKAGE_ID             3
   PACKAGE_NAME           ORA700kge_20160731211334
   PACKAGE_DESCRIPTION
   DRIVING_PROBLEM        2
   DRIVING_PROBLEM_KEY    ORA 700 [kgerev1]
   DRIVING_INCIDENT       42605
   DRIVING_INCIDENT_TIME  N/A
   STATUS                 Generated (4)
   CORRELATION_LEVEL      Typical (2)
   PROBLEMS               2 main problems, 0 correlated problems
   INCIDENTS              2 main incidents, 0 correlated incidents
   INCLUDED_FILES         84

   PACKAGE_ID             4
   PACKAGE_NAME           IPSPKG_20160801203518
   PACKAGE_DESCRIPTION
   DRIVING_PROBLEM        N/A
   DRIVING_PROBLEM_KEY    N/A
   DRIVING_INCIDENT       N/A
   DRIVING_INCIDENT_TIME  N/A
   STATUS                 Generated (4)
   CORRELATION_LEVEL      Typical (2)
   PROBLEMS               0 main problems, 0 correlated problems
   INCIDENTS              0 main incidents, 0 correlated incidents
   INCLUDED_FILES         27


$ tfactl ips show package 4 detail

Multiple ADR basepaths were found, please select one ...

( ) option[0] /scratch/app/oradb
( ) option[1] /scratch/app/oragrid

Pls select an ADR basepath [0..1] ?0
/scratch/app/oradb was selected


Multiple ADR homepaths were found for /scratch/app/oradb, please select
one ...

( ) option[0] diag/rdbms/racone/racone_2
( ) option[1] diag/rdbms/rdb11204/rdb112041
( ) option[2] diag/rdbms/ogg11204/ogg112041
( ) option[3] diag/rdbms/apxcmupg/apxcmupg_1
( ) option[4] diag/rdbms/apxcmupg/apxcmupg_2
    option[5] Done

Pls select a homepath [5] ?1
diag/rdbms/rdb11204/rdb112041 was selected

DETAILS FOR PACKAGE 4:
   PACKAGE_ID             4
   PACKAGE_NAME           IPSPKG_20160801203518
   PACKAGE_DESCRIPTION
   DRIVING_PROBLEM        N/A
   DRIVING_PROBLEM_KEY    N/A
   DRIVING_INCIDENT       N/A
   DRIVING_INCIDENT_TIME  N/A
   STATUS                 Generated (4)
   CORRELATION_LEVEL      Typical (2)
   PROBLEMS               0 main problems, 0 correlated problems
   INCIDENTS              0 main incidents, 0 correlated incidents
   INCLUDED_FILES         27
   SEQUENCES              Last 1, last full 1, last base 0
   UNPACKED               FALSE
   CREATE_TIME              2016-08-01 20:35:18.684231 -07:00
   UPDATE_TIME              N/A
   BEGIN_TIME               2016-08-01 13:59:04.000000 -07:00
   END_TIME                 2016-08-01 20:34:50.000000 -07:00
   FLAGS                    0

HISTORY FOR PACKAGE 4:
   SEQUENCE                 1
   BASE_SEQUENCE            1
   MODE                     Complete (0)
   STATUS                   Generated (4)
   FILENAME                 /scratch/app/oragrid/tfa/repository/suptools/srdc/
user_oradb/IPSPKG_20160801203518_COM_1.zip
   ARCHIVE_TIME             2016-08-01 20:35:21.899095 -07:00
   UPLOAD_TIME              N/A
   UNPACK_TIME              N/A
   FORCE                    FALSE
   GENERATE_FLAGS           0
   UNPACK_FLAGS             0

MAIN INCIDENTS FOR PACKAGE 4:
CORRELATED INCIDENTS FOR PACKAGE 4:

FILES FOR PACKAGE 4:
   FILE_ID                  1
   FILE_LOCATION            <ADR_HOME>/trace
   FILE_NAME                alert_rdb112041.log
   LAST_SEQUENCE            1
   EXCLUDE                  Included

   FILE_ID                  2087
   FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
   FILE_NAME                IPS_CONFIGURATION.dmp
   LAST_SEQUENCE            1
   EXCLUDE                  Included

   FILE_ID                  2088
   FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
   FILE_NAME                IPS_PACKAGE.dmp
   LAST_SEQUENCE            1
   EXCLUDE                  Included

   FILE_ID                  2089
   FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
   FILE_NAME                IPS_PACKAGE_INCIDENT.dmp
   LAST_SEQUENCE            1
   EXCLUDE                  Included

   FILE_ID                  2090
   FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
   FILE_NAME                IPS_PACKAGE_FILE.dmp
   LAST_SEQUENCE            1
   EXCLUDE                  Included

   FILE_ID                  2091
   FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
```

```
FILE_NAME                IPS_PACKAGE_HISTORY.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2092
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                IPS_FILE_METADATA.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2093
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                IPS_FILE_COPY_LOG.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2094
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_ACTION_DEF.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2095
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_ACTION_PARAMETER_DEF.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2096
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_ACTION.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2097
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_ACTION_PARAMETER.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2098
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_INCIDENT_TYPE.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2099
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                DDE_USER_INCIDENT_ACTION_MAP.dmp
LAST_SEQUENCE            1
EXCLUDE                  Included

FILE_ID                  2100
FILE_LOCATION            <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME                INCIDENT.dmp
LAST_SEQUENCE            1
```

```
EXCLUDE                Included

FILE_ID                2101
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME              INCCKEY.dmp
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2102
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME              INCIDENT_FILE.dmp
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2103
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME              PROBLEM.dmp
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2104
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME              HM_RUN.dmp
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2105
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/export
FILE_NAME              EM_USER_ACTIVITY.dmp
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2106
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1
FILE_NAME              config.xml
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2107
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/opatch
FILE_NAME              opatch.log
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2108
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1/opatch
FILE_NAME              opatch.xml
LAST_SEQUENCE          1
EXCLUDE                Included

FILE_ID                2109
FILE_LOCATION          <ADR_HOME>/incpkg/pkg_4/seq_1
FILE_NAME              metadata.xml
LAST_SEQUENCE          1
EXCLUDE                Included
```

```
FILE_ID                 2110
FILE_LOCATION           <ADR_HOME>/incpkg/pkg_4/seq_1
FILE_NAME               manifest_4_1.xml
LAST_SEQUENCE           1
EXCLUDE                 Included

FILE_ID                 2111
FILE_LOCATION           <ADR_HOME>/incpkg/pkg_4/seq_1
FILE_NAME               manifest_4_1.html
LAST_SEQUENCE           1
EXCLUDE                 Included

FILE_ID                 2112
FILE_LOCATION           <ADR_HOME>/incpkg/pkg_4/seq_1
FILE_NAME               manifest_4_1.txt
LAST_SEQUENCE           1
EXCLUDE                 Included
```

E.4.6.23 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]
```

Parameters

Table E-55 tfactl ips UNPACK FILE Command Parameters

Parameter	Description
file_spec	Specify the file with file name and full path.
path	Specify the path where the physical package file should be unpacked.

Example E-47 tfactl ips UNPACK FILE

```
$ tfactl ips unpack file /tmp/IPSPKG_20061026010203_COM_1.zip into /tmp/newadr
```

E.4.6.24 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]
```

Parameters

Table E-56   tfactl ips UNPACK PACKAGE Command Parameters

Parameter	Description
pkg_name	Specify the name of the package.
path	Specify the path where the physical package files should be unpacked.

Example E-48   tfactl ips UNPACK PACKAGE

```
$ tfactl ips unpack package IPSPKG_20061026010203 into /tmp/newadr
```

E.4.6.25 tfactl ips USE REMOTE KEYS

Use the `tfactl ips USE REMOTE KEYS` 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
```

Parameters

Table E-57   tfactl ips USE REMOTE KEYS Command Parameters

Parameter	Description
file_spec	Specify the file with file name and full path.
package_id	Specify the ID of the package to add the incidents to.

Example E-49   tfactl ips USE REMOTE KEYS

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

E.4.7 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 E-58 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 E-59 tfactl managelogs Show Options

Show Option	Description
`-older`	Time period for change in log volume.
`-gi`	Space utilization under `GIBASE`.
`-database`	Space utilization for Oracle database logs (Default is all, else provide a list).

E.4.8 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 n[h|d] [-force]
```

Example E-50 tfactl purge

To remove file(s) older than 30 days:

```
$ tfactl purge -older 30d
```

To remove file(s) older than 10 hours:

```
$ tfactl purge -older 10h
```

F

Behavior Changes, Deprecated and Desupported Features

Review information about changes, deprecations, and desupports.

- Oracle Database Quality of Service (QoS) Management is Deprecated and Desupported in Release 21c
  Starting in Oracle Database release 21c, Oracle Database Quality of Service (QoS) Management is deprecated and desupported.

F.1 Oracle Database Quality of Service (QoS) Management is Deprecated and Desupported in Release 21c

Starting in Oracle Database release 21c, Oracle Database Quality of Service (QoS) Management is deprecated and desupported.

Oracle Database Quality of Service (QoS) Management automates the workload management for an entire system by adjusting the system configuration based on pre-defined policies to keep applications running at the performance levels needed. Applications and databases are increasingly deployed in systems that provide some of the resource management capabilities of Oracle Database Quality of Service (QoS) Management. At the same time, Oracle's Autonomous Health Framework has been enhanced to adjust and provide recommendations to mitigate events and conditions that impact the health and operational capability of a system and its associated components. For those reasons, Oracle Database Quality of Service (QoS) Management has been deprecated and desupported with Oracle Database 21c.