Oracle® Enterprise Manager

Getting Started with the Oracle Diagnostics Pack

Release 2.1

February 2000

Part No. A76915-01

ORACLE®

Getting Started with the Oracle Diagnostics Pack, Release 2.1

Part No. A76915-01

Primary Author: Peter LaQuerre

Contributing Authors: Jackie Gosselin and Bert Rich

Contents

3 Introduction to Oracle Performance Manager

4 Introduction to Oracle Capacity Planner

5 Using Oracle TopSessions

6 Oracle Advanced Event Tests

Send Us Your Comments

Getting Started with the Oracle Diagnostics Pack, Release 2.1

Part No. A76915-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail - nedc_doc@us.oracle.com
- FAX - 603-897-3317. Attn: Oracle Diagnostics Pack Documentation
- Postal service:
  Oracle Corporation
  Oracle Diagnostics Pack Documentation
  98 Spit Brook Road - Suite 300
  Nashua, NH 03062
  U.S.A.

If you would like a reply, please give your name, address, and telephone number below.

If you have problems with the software, please contact your local Oracle World Wide Support Center.

Preface

This section describes the purpose and organization of this guide: *Getting Started with the Oracle Diagnostics Pack*. Specifically, it covers the following topics:

Purpose of This Guide

This guide provides an overview of the Oracle Performance Manager, Oracle Capacity Planner, and Oracle TopSessions applications and their features. It also describes the Oracle Event Tests that are provided with the Oracle Diagnostics Pack. As the guide describes how to use these applications and Oracle Event Tests provided with the Oracle Diagnostics Pack, it also introduces you to key concepts and terminology.

Audience

This guide is written for those who wish to use Oracle Performance Manager, Oracle Capacity Planner, Oracle TopSessions, and the Oracle Event Tests that are provided with the Oracle Diagnostics Pack.

This guide assumes that you are familiar with the Oracle Enterprise Manager console and base applications. If you are not, please read the *Oracle Enterprise Manager Concepts Guide* and the *Oracle Enterprise Manager Administrator's Guide*.

How This Guide Is Organized

This guide is organized as follows:

Chapter 1, "Overview"

Provides an overview of the Oracle Diagnostics Pack and its features.

Chapter 2, "Post-Installation Configuration"

Provides lists of steps to consider after the installation of the Oracle Diagnostics Pack has taken place.

Chapter 3, "Introduction to Oracle Performance Manager"

Provides an overview of the Oracle Performance Manager application and its features.

Chapter 4, "Introduction to Oracle Capacity Planner"

Provides an overview of the Oracle Capacity Planner application and its features.

Chapter 5, "Using Oracle TopSessions"

Provides an overview of the Oracle TopSessions application and its features.

Chapter 6, "Oracle Advanced Event Tests"

Provides an introduction to the Oracle Event Tests that are provided with the Oracle Diagnostics Pack.

Appendix A, "Using Advanced Event Tests Effectively"

Provides several case studies demonstrating how the Oracle Event Tests provided with the Oracle Diagnostics Pack can help database administrators simplify daily administration tasks.

How to Use This Guide

This guide should be read in its entirety to provide you with an overview of Oracle Performance Manager, Oracle Capacity Planner, Oracle TopSessions, and the Oracle Event Tests provided with the Oracle Diagnostics Pack, so that you are familiar with the tasks that the applications can perform and with the Oracle Event Tests that are

provided. The online help provides more information about the applications and the Oracle Event Tests provided with the Oracle Diagnostics Pack.

Oracle Enterprise Manager Pack Information

The Oracle Enterprise Manager product family includes six packs: Oracle Change Management Pack, Oracle Diagnostics Pack, Oracle Tuning Pack, Oracle Standard Management Pack, Oracle Management Pack for SAP R/3, and Oracle Management Pack for Oracle Applications. Each pack is fully integrated into the Oracle Enterprise Manager Console framework.

Oracle Change Management Pack

- Includes Baseline Viewer, Change Manager, DB Alter, DB Capture, DB Diff, DB Propagate, DB Quick Change, DB Search, and Plan Editor.
- Tracks metadata changes in databases.
- Eliminates errors and loss of data when upgrading databases to support new applications.
- Analyzes the impact and complex dependencies associated with application change and automatically performs database upgrades.
- Allows you to find the database objects that match a set of search criteria that you specify.
- Initiates change with easy-to-use wizards that teach systematic steps necessary to upgrade databases.

Oracle Diagnostics Pack

- Includes Oracle Performance Manager, Oracle Capacity Planner, Oracle Trace, Oracle TopSessions, and Oracle Advanced Event Tests.
- Monitors, diagnoses, and maintains the stability of databases, operating systems, and applications. Both historical and real-time analyses are used to automatically avoid problems before they occur.
- Provides powerful capacity planning features that enable users to easily plan and track future system resource requirements.

Oracle Tuning Pack

- Includes Oracle SQL Analyze, Oracle Expert, Oracle Index Tuning Wizard, Reorg Wizard, and the Tablespace Map.

- Optimizes system performance by identifying and tuning major database and application bottlenecks such as inefficient SQL, poor data structures, and improper use of system resources.
- Discovers tuning opportunities and automatically generates the analysis and required changes to tune the system. Inherent in the product are powerful teaching tools that help DBAs learn to tune the system as they work.
- Helps increase developers and DBAs productivity.

Oracle Standard Management Pack

- Includes Baseline Viewer, Change Manager, DB Capture, DB Diff, Oracle Index Tuning Wizard, Oracle Performance Manager, and Oracle TopSessions.
- Monitors and diagnoses problems, tunes high impact indexes, and tracks and compares changes in your Oracle environment.

Oracle Management Pack for SAP R/3

- Includes Oracle Performance Manager, Oracle Capacity Planner, and Oracle Event Tests that are specific to monitoring your SAP R/3 environment.
- Monitors, diagnoses, and maintains the health of a SAP R/3 system. Both historical and real-time analysis are used to automatically avoid problems before they occur.
- Provides powerful capacity planning features that enable users to easily plan and track future system resource requirements.

Oracle Management Pack for Oracle Applications

- Includes Oracle Performance Manager, Oracle Capacity Planner, Concurrent Processing Tuning Assistant, and Oracle Applications Event Tests.
- Enables you to monitor all aspects of your system, including databases and concurrent managers.
- Enables the monitoring, diagnosing, and capacity planning of the Oracle Applications environment.

Oracle Enterprise Manager Documentation

Getting Started with the Oracle Diagnostics Pack is one of several Oracle Enterprise Manager documents.

Oracle Enterprise Manager Base Documentation

- *Oracle Enterprise Manager Readme* provides important notes regarding the online documentation, updates to the software, and other late-breaking information.
- *Oracle Enterprise Manager Administrator's Guide* explains how to use Oracle Enterprise Manager, the Oracle systems management console, common services, and integrated platform tool.
- *Oracle Enterprise Manager Concepts Guide* provides an overview of Oracle Enterprise Manager.
- *Oracle Enterprise Manager Configuration Guide* explains how to configure Oracle Enterprise Manager.
- *Oracle Enterprise Manager Messages Manual* describes Oracle Enterprise Manager error messages and methods for diagnosing the errors.
- *Oracle Intelligent Agent User's Guide* provides configuration information and answers to crucial troubleshooting questions pertaining to the Oracle Intelligent Agent.

Oracle Change Management Pack Documentation

- *Oracle Change Management Pack Readme* provides important notes regarding the Change Management Pack online documentation, updates to the software, and other late-breaking information.
- *Oracle Enterprise Manager Getting Started with Oracle Change Management Pack* provides an overview of the concepts and features of Oracle Change Management Pack applications.

Oracle Diagnostics Pack Documentation

- *Oracle Diagnostics Pack Readme* provides important notes regarding Diagnostics Pack online documentation, updates to the software, and other late-breaking information.
- *Getting Started with the Oracle Diagnostics Pack* provides an overview of the concepts and features of Oracle Performance Manager, Oracle Capacity Planner, and Oracle TopSessions applications. It also describes Oracle Event Tests that are provided with the Oracle Diagnostics Pack, which allow you to implement automatic problem detection and correction on databases, listeners, and nodes.
- *Oracle Enterprise Manager Oracle Trace User's Guide* explains how to use Oracle Trace to capture and use historical data to monitor Oracle databases.

Oracle Tuning Pack Documentation

- *Oracle Tuning Pack Readme* provides important notes regarding Tuning Pack online documentation, updates to the software, and other late-breaking information.
- *Oracle Enterprise Manager Database Tuning with the Oracle Tuning Pack* provides an overview of the concepts and features of each of the applications included in the Oracle Tuning Pack. The applications include: Oracle SQL Analyze, Oracle Expert, Oracle Index Tuning Wizard, Reorg Wizard, and the Tablespace Map. A description of how these applications can work together to tune an Oracle database is also provided.

Oracle Standard Management Pack Documentation

- The *Oracle Standard Management Pack Readme* provides important notes regarding the Oracle Standard Management Pack online documentation, updates to the software, and other late-breaking information.
- The *Getting Started with Oracle Standard Management Pack* manual provides an overview of the concepts and features of Baseline Viewer, Change Manager, DB Capture, DB Diff, Oracle Index Tuning Wizard, Oracle Performance Manager, and Oracle TopSessions.

Oracle Management Pack for SAP/R3 Documentation

- *Oracle Management Pack for SAP/R3 Readme* provides important notes regarding Oracle Management Pack for SAP R/3 online documentation, updates to the software, and other late-breaking information.
- *Getting Started with the Oracle Management Pack for SAP/R3* provides an overview of the concepts and features of Oracle Performance Manager and Oracle Capacity Planner. It also describes Oracle Event Tests that are provided with the Oracle Management Pack for SAP/R3.

Oracle Management Pack for Oracle Applications Documentation

- *Oracle Management Pack for Oracle Applications Readme* provides important notes regarding Oracle Management Pack for Oracle Applications online documentation, updates to the software, and other late-breaking information.
- *Getting Started with the Oracle Management Pack for Oracle Applications* provides an overview of the concepts and features of Oracle Performance Manager, Oracle Capacity Planner, and Concurrent Processing Tuning Assistant. It also describes Oracle Applications Event Tests and Jobs that are specific to monitoring your Oracle Applications environment.

- *Oracle Intelligent Agent User's Guide* provides configuration information and answers to crucial troubleshooting questions pertaining to the Oracle Intelligent Agent.

1

Overview

The Oracle Diagnostics Pack extends Oracle Enterprise Manager to enable the monitoring, diagnosing, and capacity planning of the multitiered Oracle server environment. The Diagnostics Pack takes advantage of the following Enterprise Manager system management features:

- Discovery and graphical representation of services, such as databases or nodes, to be monitored
- Starting tools from the console in the context of a specific service
- Automated collection of performance and resource usage data
- Central monitoring and administration of remote systems using intelligent agents

The Diagnostics Pack provides the following components:

- Oracle Advanced Event Tests

  A set of predefined event tests built on the Oracle Event Management System offering "lights out" performance monitoring and problem solving for distributed systems.

- Oracle Performance Manager

  A real-time performance monitor, providing graphical views of an array of performance metrics for the services you are monitoring. Performance Manager provides the ability to drill down to detailed data of performance diagnostics data to identify problems such as lock contention and excessive file I/O.

- Oracle Capacity Planner

  A tool for collecting, storing, and analyzing historical performance data for the services you are monitoring. Capacity Planner allows you to use historical

information to diagnose problems today and to anticipate resource needs to avoid problems in the future.

- Oracle Trace

  A comprehensive tracing mechanism for the Oracle Server. Oracle Trace captures SQL, transaction, and application-based workload and performance data.

- Oracle TopSessions

  A diagnostics tool that pinpoints problematic database sessions and displays detailed performance and resource consumption data.

This chapter provides a brief overview of each of these components and gives a brief example of how to best use these components to solve database performance problems.

Oracle Advanced Event Tests

In addition to the basic UpDown event tests provided for all services administered in the Enterprise Manager console, the Diagnostics Pack provides a library of advanced event tests to automatically detect excessive resource use, performance degradation, and other problem situations.

See Chapter 6, "Oracle Advanced Event Tests" for a complete list of these advanced event tests.

The key to a smooth running system is to resolve problems before they turn into crises. These advanced event tests allow you to focus on the causes of a problem before the symptoms become serious. In addition, with the paging and E-mail support provided by Enterprise Manager, you can be notified of any problems with the system wherever you happen to be.

Oracle Performance Manager

Performance Manager displays performance data in real-time graphical views that can be automatically refreshed at user-defined intervals. Multiple charts and tables can be displayed simultaneously, affording you a multifaceted view of system performance. For example, an Oracle server administrator can monitor resources consumed by the database, the underlying operating system, and other processes running on the node. Some of the predefined overview charts are:

- Free space available on selected disks

- Memory usage on selected nodes
- Cache utilization
- Throughput

Users can further drill down on these charts to see greater detail, and in some cases, drill down to problematic sessions and terminate them. Many other subsystem specific charts are available, giving you a rich set of data to analyze your system from multiple viewpoints.

Oracle Capacity Planner

The same data collection mechanism used by Performance Manager to sample real-time data is also used to collect historical performance and resource consumption data. The Diagnostics Pack uses Capacity Planner to analyze performance data to help you diagnose problems using historical information and project your future capacity needs.

In Performance Manager, historical data can be gathered for the services you monitor. The collection can be customized, allowing you to select any subset of statistics to collect and the interval at which these statistics should be sampled. The data collection mechanism provides roll up capabilities, where data is automatically aggregated at the end of each hour, day, week, or month. This data collection mechanism also allows you to set up data expiration policies so you can fully control the amount of storage needed for the historical data.

Once the data is gathered, you can use Capacity Planner to chart the historical data for the purposes of identifying trends and predicting future hardware requirements for the system. You can also extrapolate to a particular point in time or work back from a particular target value and find when a specific level of usage is expected to occur. These charts can be customized, allowing you to get the data you need and to view data from multiple sources (for example, CPU and disk) on a single, unified chart.

Oracle Trace

Trace is a general-purpose tracing mechanism that collects data for any software product enabled with Trace Application Programming Interface (API) calls, such as the Oracle server. You can use Trace to collect a wide variety of data, such as performance statistics, diagnostic data, system resource usage, and business transaction details.

The major components of Trace are the Trace Manager and Trace Data Viewer. Trace Manager is a graphical application used to create, schedule and administer Trace collections for products containing Trace API calls. Trace Manager automatically discovers Trace preconfigured products that are installed on all nodes that are known to the Oracle Management Server. By default Trace discovers any Oracle Server release 7.3.3 and higher.

Data Viewer allows you to view formatted data collected by Trace. Data Viewer handles the complex task of extracting data and aggregating key server performance metrics on a large Trace collection. Once you select a Trace collection, you can have Data Viewer compute SQL or Wait statistics or both.

Refer to the *Oracle Trace User's Guide* for additional information about Trace.

Oracle TopSessions

TopSessions is a tool for monitoring how connected sessions use database instance resources in real time. You can obtain an overview of session activity, by displaying the top "n" sessions sorted by a statistic of your choosing. For any given session, you can then drill down for more detail. You can further customize the information you display by specifying manual or automatic data refresh, the rate of automatic refresh, and the number of sessions to display.

In addition to these useful monitoring capabilities, TopSessions provides a methodology for identifying and correcting certain database performance problems. For example, when sudden file I/O load is detected, you can first identify the sessions contributing most to the problem, and then isolate the executing SQL statements in user applications for those sessions. You can then analyze the SQL explain plans for those SQL statements to determine how best to resolve the problem.

Typical Scenario for Using the Diagnostics Pack

Monitoring, problem diagnosis, and correction can be broken down into three stages:

- Detecting a problem
- Diagnosing the cause of the problem
- Correcting the problem and preventing the problem from occurring again

The following sections provide an example of how you can use the Diagnostics Pack to solve performance problems.

System Monitoring and Problem Detection

Problems can come to your attention in a number of ways:

- You are alerted by your users that there is a problem, for example, the system is very slow
- You notice something yourself
- You get an alarm, page, or E-mail that the system is not working correctly or that a potential problem may exist

Ideally an administrator would like to be alerted to a problem before his or her users are aware of the problem. The Advanced Event Tests included in the Diagnostics Pack allow you to proactively identify potential problems before they become serious and adversely affect users.

Diagnosing the Cause of the Problem

You can diagnose the cause of the problem by:

- Examining the current state of the system
- Investigating resource usage
- Identifying problem sessions
- Identifying high impact SQL
- Analyzing past (historical) information

Examining the Current State of the System

Using Performance Manager, you can take a quick look at the system resources by accessing the Overview chart. This chart provides the following information:

- Processor, user, and privileged times across all processors
- Bytes available on each disk
- Current processor queue length
- Available and committed memory
- Counts of various objects on the system
- System I/O operations

Investigating Resource Usage

Using Performance Manager you can identify the processes using the most resources. The Process Info Chart found under the Nodes branch of the navigator tree, lists the Percent Memory Used. The Memory Statistics Chart found under the Databases branch of the navigator tree, lists memory statistics, for example, Shared Pool Size, Buffer Cache Size, and SGA Size.

Identifying Problem Sessions

Using Oracle TopSessions you can identify the top *n* sessions using the most CPU, memory, and I/O. You can also drill down into sessions statistics, for example, V$: sesstat, open_cursor, locks, and sqlarea.

You can also sort data based on a selected statistics filter. For example, you can sort the top 10 sessions for the Redo activity by redo blocks written, redo size, and redo entry volume.

Finally, you can view open cursors, explain plans, and sessions holding locks that other sessions are waiting for.

Identifying High Impact SQL

Using SQL Analyze, an application available in the Oracle Tuning Pack, you can select and sort the top *n* cached SQL statements based on the SQL volume metrics and SQL efficiency metrics. The SQL volume metrics include: number of executions, disk reads, and sorts. The SQL efficiency metrics include: buffer gets per executions, buffer gets per rows processed, and parse calls per executions.

Also using SQL Analyze, you can drag and drop SQL statements to the SQL Analyze workbench for tuning.

Analyzing Past Information

If a collection was set using Capacity Planner, you can determine what led up to the problem. For example, you can collect data to determine space usage and answer the following questions:

- What led up to the space problem? Was the space consumed gradually or all at once?
- Does the space problem still exist?
- Is the space problem getting any better?

By using the trend analysis feature of Capacity Planner, you can determine when the disk will be full.

Correcting the Problem

Once you have diagnosed the problem, there are various actions you can take to resolve the problem. Some examples include:

- Starting fixit jobs in the Enterprise Manager console

  Define an event to automatically run a fixit job when the event is triggered.

- Identifying and correcting resource-intensive sessions

  Either kill or block a session if the session is consuming too many resources.

- Identifying problematic SQL statements

  Use Performance Manager to drill down to a SQL chart and launch SQL Analyze to tune the problematic SQL statement.

- Analyzing space usage

  Add a datafile in a tablespace storage folder. You can also change the MAXEXTENTS of a table.

- Improving memory usage

  Analyze the buffer size DB_BLOCK_BUFFER parameter. You can also change the initialization parameters using the Instance Manager application.

- Tuning the database by using the Oracle Tuning Pack
- Reorganizing tablespaces using the Oracle Tuning Pack

2

Post-Installation Configuration

After you have installed the Diagnostics Pack, configure the components listed in the following table:

Components	Where to Find the Information
Advanced Events	"Setting Up Advanced Events" on page 2-1
Capacity Planner	"Setting Up Capacity Planner and Performance Manager" on page 2-2
Performance Manager	"Setting Up Capacity Planner and Performance Manager" on page 2-2
TopSessions	"Setting Up TopSessions" on page 2-5
Trace	"Setting Up Trace" on page 2-7

Setting Up Advanced Events

Before you use Oracle Advanced Events, set up the program as follows:

- Create additional tables, views and public synonyms required for the Advanced Events.

  Refer to "Setting Up TopSessions" on page 2-5 for information on what scripts to run.

- If you plan to use the Chained Rows advanced events, the CHAINED_ROWS table must exist in the database. To create the CHAINED_ROWS table:

  1. Log into the database as SYS.
  2. Run the utlchain.sql script. This script is located in the $ORACLE_HOME/RDBMS/ADMIN directory.

Setting Up Capacity Planner and Performance Manager

The following setup tasks are required before you use Oracle Capacity Planner or Oracle Performance Manager:

- ❏ Configure the Oracle Data Gatherer to collect data
- ❏ Optionally, enable disk statistics collection on Windows NT

The following setup tasks are required before you use Performance Manager:

- ❏ Convert any user-defined charts you created with earlier versions of Performance Manager
- ❏ If you plan to use Performance Manager in a Parallel Server environment, you may need to set up your server using Oracle-supplied SQL scripts.

The following sections describe each of these setup tasks in more detail.

Configuring Oracle Data Gatherer to Collect Data

Performance Manager and Capacity Planner require the use of the data gathering service (also known as the Oracle Data Gatherer). For monitoring operating system data, the Oracle Data Gatherer must be installed on the same node as the operating system you are monitoring. For monitoring other service types, such as databases, the Oracle Data Gatherer can run on the same node as the service you are monitoring, or it can run on a different node, depending upon your configuration.

Refer to the *Oracle Intelligent Agent User's Guide* for information on how to configure and start the Oracle Data Gatherer.

Enabling Disk Statistics Collection on Windows NT

To collect disk statistics on Windows NT, enable them by typing:

```
diskperf  -Y
```

Disk statistics are enabled the next time the system is restarted. For more information about the DISKPERF tool, refer to the Microsoft Windows NT documentation.

Note: Enabling disk statistics is not required. You can still monitor other operating system statistics with Performance Manager or Capacity Planner even if disk statistics are not enabled.

Converting User-Defined Charts From Previous Releases of Performance Manager

If you created user-defined charts using Performance Manager release 1.5.0 or earlier and want to use them with the current Performance Manager release, you must first convert them to release 1.6.0.

Performance Manager Release 1.6.0 user-defined charts are converted to the current release when you migrate your existing performance data to the Enterprise Manager Release 2.1 repository.

For more information, refer to the following sections.

Note: Release 1.5.5 of Performance Manager did not include the functionality of user-defined charts.

Converting Performance Manager Release 1.5.0 User-Defined Charts

To convert user-defined charts created using Performance Manager Release 1.5.0 or earlier:

1. Run vmmmig.exe. This executable creates a text file named vtmusr.txt in the $ORACLE_HOME\SYSMAN\ADMIN directory. This text file contains data about the user-defined charts created using Performance Manager release 1.5.0 or earlier. When you run vmmmig.exe, specify the user name, password, and service for the Enterprise Manager repository that contains the user-defined charts that you want to convert, for example:

   ```
   vmmmig joseph/password@my_rep
   ```

   Note that `my_rep` in the previous command line is the service name for the Enterprise Manager repository.

2. Run vmm2vtm.exe. This executable uses the data in the vtmusr.txt file to create user-defined charts in the repository that can be used with Performance Manager release 1.6.0. When you run vmm2vtm.exe, supply a user name, password, and service for the Enterprise Manager V2.x repository and the name of the service under which you want the user-defined charts stored for Performance Manager release 1.6.0, for example:

   ```
   vmm2vtm.exe joseph/password@my_rep my_db
   ```

   Note that my_rep in the previous command line is the service name of the Enterprise Manager V2.x repository and my_db is the name of the target service under which you want the user-defined charts stored. In other words, after the

previous command is run, all the user-defined charts are converted and stored under the my_db service in the Performance Manager release 1.6.0 tree view.

If any error messages are generated when you run vmm2vtm.exe, edit the vtmusr.txt file based on the error messages. For example, the error messages may state that there is a problem with a chart or that a chart does not exist. Since the vtmusr.txt file contains a list of the charts, edit the file to remove any problematic charts.

Then, run vmm2vtm.exe again.

Converting Performance Manager Release 1.6.0 User-Defined Charts

If you created user-defined charts with Performance Manager Release 1.6.0, you can convert those charts for use with the current release of Performance Manager. User-defined charts created with Performance Manager 1.6.0 are converted when you create a new Enterprise Manager repository and migrate your existing performance data to the new repository.

For more information about creating a new Oracle Enterprise Manager repository and migrating your existing data, see the *Oracle Enterprise Manager Configuration Guide*.

Using Performance Manager in a Parallel Server Environment

To use Performance Manager in a Parallel Server environment, two database instances must be started, otherwise Performance Manager does not treat the server as a Parallel Server.

There are two scenarios for using Performance Manager in an Oracle Parallel Server environment, and different configuration steps for each scenario.

The two scenarios and their configuration steps are:

1. If you are using Performance Manager to monitor an Oracle7 Parallel Server environment, do the following:
   a. Ensure all instances are running.
   b. Run the ops_gdl.sql script. This generates an appropriate ops_dbl.sql file. The ops_dbl.sql file shipped with the product is only an example. The actual contents of ops_dbl.sql depend on your database name, instance names, and the number of instances which exist.

c. Run the ops_mon.sql script, which runs ops_ctab.sql, ops_dbl.sql, ops_pack.sql to create Performance Manager tables and views to fetch and display Oracle7 Parallel Server data in charts.

   Oracle7 does not have global V$ (GV$) tables; these files install a set of PL/SQL procedures, database links, and O$ tables that are used by Performance Manager in the absence of GV$ tables.

2. If you are using Performance Manager to monitor an Oracle8 Parallel Server environment, you do not need to run any of the Parallel Server scripts. For example, if you use any release of Oracle that has GV$ tables (namely Oracle release 8.0 and later), you do not need to run these scripts.

For more information about Parallel Server scripts, see the *Oracle Parallel Server Management User's Guide* in the Parallel Server documentation set.

Setting Up TopSessions

To install the features of TopSessions:

- ❏ Create additional tables and views on each database you want to connect to from TopSessions
- ❏ Grant certain SELECT privileges

The smptsi*xx*.sql scripts have been provided to help automate this process. The *xx* in the file name identifies the version of the database against which the script should be run. The script for each database version is located in the $ORACLE_HOME\SYSMAN\ADMIN directory.

Version of the Database	Script to Run
Oracle 7.3.4	smptsi73.sql script
Oracle 8.0	smptsi80.sql script
Oracle 8.1	smptsi81.sql script

When smptsi*xx*.sql is run, it also automatically runs the following two scripts:

- catblo*xx*.sql
- utlxpl*xx*.sql

These two scripts create in the managed database some additional tables, views, and public synonyms that are required by the Oracle Advanced Events.

To set up TopSessions for a database:

1. Access SQL*Plus Worksheet from the Start menu by following the path: Start=>Programs=>ORACLE_HOME=>DBA Management Pack=>SQLPlus Worksheet.

   Note: ORACLE_HOME represents the Oracle home in which the Diagnostics Pack is installed.

2. Use the Oracle Enterprise Manager Login dialog box to connect to the managed database as SYS.

   For additional information, refer to the online help or the *Oracle Enterprise Manager Administrator's Guide*.

3. Grant `SELECT ANY TABLE` privileges to each administrator account. This step may be omitted if the account has already been granted the same privileges as SYSTEM.

   Note that when preparing to run the smptsi*xx*.sql script on managed databases, you should log into each database as SYS, as mentioned previously.

   You can use Security Manager to grant privileges to an account. Refer to the Oracle Enterprise Manager Online Help for detailed information about how to use Security Manager.

4. Run the smptsi*xx*.sql script for the managed database.

   If the smptsi*xx*.sql script is not run on a managed database, you may see a "Table or View does not exist" message when you try to use TopSessions.

5. If the ALL_CATALOG view does not exist on the managed database, run the catalog.sql script on the database from the SYS account. The catalog.sql script is located in the $ORACLE_HOME\RDBMSxx\ADMIN directory.

6. If the AUDIT_ACTIONS view does not exist on the managed database, run the cataudit.sql script on the database from the SYS account. The cataudit.sql script is located in the $ORACLE_HOME\RDBMSxx\ADMIN directory.

7. Exit SQL*Plus Worksheet.

Setting Up Trace

Trace requires that the:

- Oracle Intelligent Agent for Oracle release 7.3.4 or later is installed and running on the nodes that are targets for Trace collections
- Oracle Management Server (OMS) is running, and the nodes on which you want to use Trace have been discovered using the Oracle Enterprise Manager Console.

To successfully use Trace, you must:

- Configure Trace for Oracle8 Server collections
- Configure Trace for Oracle 7.3.4 Server collections
- Create Trace Formatter Tables

Using Trace for Oracle8 Server Collections

If you are using Trace for Oracle8 server collections, you must set the value of the ORACLE_TRACE_ENABLE parameter in your INIT*sid*.ORA file to TRUE.

Using Trace for Oracle 7.3.4 Server Collections

If you are using Trace to collect data from Oracle 7.3.4 server collections, verify the following exist:

- Trace user account: TRACESVR
- Trace stored procedure packages: DBMS_ORACLE_TRACE_AGENT and DBMS_ORACLE_TRACE_USER

If they do not, you must create them by running the otrcsvr.sql script as SYS.

The otrcsvr.sql script is located in $ORACLE_HOME/otrace/admin on UNIX systems and in $ORACLE_HOME\otracexx\admin on Windows NT systems.

The otrcsvr.sql script is run automatically during database installation on most platforms. However, if your server platform is Windows NT, you must run this script manually.

Creating Trace Formatter Tables

If you are using the Trace Collection Services release 8.0.4 or later, the formatter tables are created for you.

The Trace formatter tables are required in managed databases by Trace, which converts and loads a Trace binary file (*collection_name*.dat) into Oracle tables for access.

Trace data collected from Oracle 7.3.4 and later databases can only be stored in databases that are using the latest Trace formatter tables. Use the vobsh command, which is available only on Windows platforms, to create new formatter tables, delete existing formatter tables, or upgrade earlier versions of formatter tables to the latest version for a database.

The formatter table operations and vobsh commands to perform these operations are described in the following sections. These vobsh commands are unsuccessful if the specified user does not have an account for the specified service. In this case, create an account for the user on the specified service, then enter the vobsh command again.

Creating New Formatter Tables

To create new formatter tables in a database where no formatter tables currently exist, use the vobsh command:

```
vobsh -c "user/password@service" -o CREATE -p "EPCFMT"
```

Note that the formatted data stored in formatter tables can grow very large, so the database in which formatter tables are created should have sufficient space to store large amounts of data.

Dropping Existing Formatter Tables

To drop existing formatter tables from a table, use the vobsh command:

```
vobsh -c "user/password@service" -o DROP -p "EPCFMT"
```

This command drops any version of formatter tables from the specified database. Note that in addition to dropping the existing formatter tables, vobsh also deletes the formatted data stored in the tables.

Upgrading Formatter Tables

To validate (upgrade) older versions of the formatter tables (which creates, drops, or upgrades the formatter tables as needed), use vobsh command:

```
vobsh -c "user/password@service" -o VALIDATE -p "EPCFMT"
```

This command determines what version of the formatter tables the database has and enables you to upgrade them to the new version, if necessary.

Other Configuration Information

If you experience any problems running Trace after completing the configuration instructions in this chapter, refer to the "Troubleshooting Oracle Trace" appendix of the *Oracle Enterprise Manager Oracle Trace User's Guide*.

3

Introduction to Oracle Performance Manager

Oracle Performance Manager is an Oracle Enterprise Manager application that allows you to monitor different types of real-time performance data.

Oracle Performance Manager uses the Agent data gathering service (also called the Oracle Data Gatherer) to collect performance data. To monitor some types of data, you must install the Oracle Data Gatherer on the system (or systems) where you want to monitor data. See the *Oracle Intelligent Agent User's Guide* for information on managing the Agent data gathering service on a system.

If you are using Oracle Performance Manager to monitor database performance, the Agent data gathering service can be installed on another system. See "Accessing Performance Data Through an Intermediate Host" on page 3-2 for more information on accessing data through an Agent data gathering service on another system.

The types of performance data that Oracle Performance Manager can monitor on a system depends on the products that are installed. Table 3–1 shows the types of data (or services) that Oracle Performance Manager can monitor when various Oracle products are installed.

Table 3–1 Data that Performance Manager Can Monitor when Various Oracle Products Are Installed

Oracle Product	Oracle Concurrent Manager Data	Oracle Server Data	Operating System Data	SAP R/3 System Data	Microsoft SQL Server Data
Diagnostics Pack		Yes	Yes		Yes

Table 3–1 Data that Performance Manager Can Monitor when Various Oracle Products Are Installed (Cont.)

Oracle Product	Oracle Concurrent Manager Data	Oracle Server Data	Operating System Data	SAP R/3 System Data	Microsoft SQL Server Data
Management Pack for Oracle Applications	Yes		Yes		
Management Pack for SAP R/3			Yes	Yes	
Standard Management Pack (Oracle Standard Edition only)		Yes	Yes		Yes

If additional products from Oracle or other vendors are installed, Oracle Performance Manager may be able to monitor additional types of data not shown in Table 3–1.

Accessing Performance Data Through an Intermediate Host

Oracle Performance Manager relies on the Oracle Data Gatherer to monitor a service (such as a node, a database, or a concurrent manager). Usually, the collection of the data is done on the system where the service is located. However, for some types of services Oracle Performance Manager can also monitor data on a system by using the Oracle Data Gatherer on another system (an intermediate host).

This intermediate host could be the client system where the Oracle Performance Manager is running, or any other system on the network on which the Data Gatherer is available.

There are three ways to set the location of the Oracle Data Gather. The method you use depends on how you connect to the service. Consider the following three scenarios:

- You connect to a database that has been discovered using the Enterprise Manager console and you are running Oracle Performance Manager while connected to the Oracle Management Server.

Performance Manager attempts to connect to the Data Gatherer on the host where the service is located. If this connection fails or cannot be attempted because no preferred credentials have been set for the database, you will be prompted with a dialog box containing logon credentials for the database. This dialog box also contains a field to specify the location of the Oracle Data Gatherer that will be used to collect performance data.

- You click the **Add New Service** button in the toolbar panel and manually add the service to the navigator.

  Performance Manager displays a Logon dialog box. This dialog box also contains a field to specify the location of the Data Gatherer that will be used to collect performance data.

- You select a service in the Performance Manager navigator tree and choose **Set Data Gatherer Location** from the **File** menu.

  Performance Manager displays the Database Logon dialog box for the selected service. This dialog box also contains a field to specify the location of the Data Gatherer that will be used to collect performance data.

Note: The database name (or service) shown in the Database Logon dialog box must be a service name which can be resolved on the system where the Agent data gathering service is located. A **tnsnames** entry or name server entry must be available for that service on the system specified in the **Data Gatherer Location** field.

Usage Scenarios for Oracle Performance Manager

Using Oracle Performance Manager, you can:

- Monitor one or more services concurrently.
- View the monitored data in various chart formats, including strip (line), pie, bar, table, and hierarchical.
- Set the refresh rate for a chart's data.
- View multiple charts concurrently for each monitored service.
- View multiple charts in a single window.
- Drill down from one chart to another related chart.
- Drill down from real-time chart data to historical chart data.

- Choose the collected data and data sources to display in a chart.
- Print a chart.
- Generate an HTML report for a chart.
- Display context-sensitive help for a chart, if help is available for the chart.
- Play a recording of a chart, with fast-forward and pause features.
- Create and modify user-defined charts.

Starting Oracle Performance Manager

You can start Oracle Performance Manager from the Oracle Enterprise Manager console or from the **Start** menu.

Starting Oracle Performance Manager from the Console

You can start Oracle Performance Manager from the Oracle Enterprise Manager console in any of the following ways:

- On the Oracle Enterprise Manager console, point to the **Diagnostics Pack** drawer, then click **Performance Manager**.
- On the **Tools** menu of the Oracle Enterprise Manager console, point to **Diagnostics Pack**, then click **Performance Manager**.

Note: When you start Oracle Performance Manager from the Oracle Enterprise Manager console, Oracle Performance Manager connects to the Oracle Management Server using the same credentials used to start the Oracle Enterprise Manager console.

Starting Oracle Performance Manager from the Start Menu

On the **Start** menu, follow the path: **Start**=>**Programs**=>**ORACLE_HOME**=>**Diagnostics Pack**=>**Performance Manager**.

Note: ORACLE_HOME represents the home directory in which the Diagnostics Pack is installed.

When you start Performance Manager from the **Start** menu, you are given a choice of logging in to the Oracle Management Server or running Performance Manager in standalone mode.

Running Performance Manager in Repository Mode

To run Oracle Performance Manager in repository mode:

1. Click **Login to the Oracle Management Server**.
2. Enter the appropriate connection information for your Oracle Management Server.

To select an alternate Management Server, click the **Management Server** button next to the **Management Server** drop-down list. In the resulting dialog box, you can add another node that is running Oracle Management Server software. The Oracle Management Server and its associated repository must already exist.

For information on the fields in the Management Servers dialog box, click **Help**.

3. Click **OK**.

If the repository information for the Diagnostics Pack has not been created, Oracle Performance Manager automatically updates the repository at connection time.

Note: If the repository was created using Enterprise Manager Release 2.0.4 or earlier, you must upgrade the repository or create a new repository. For more information, see the *Oracle Enterprise Manager Configuration Guide*.

Running Performance Manager in Standalone Mode

To run Oracle Performance Manager in standalone mode (not connected to the Oracle Enterprise Manager repository), click **Standalone, no repository connection**, then click **OK**.

Note: When you run Oracle Performance Manager without connecting to a valid repository, you cannot save any user-defined charts, and any recordings you make during the session will not be saved when you exit Oracle Performance Manager.

In addition, you will need to manually add services you want to monitor, and you will have to add them again each time you start Performance Manager.

Using the Oracle Performance Manager Main Window

The primary component of the Oracle Performance Manager user interface is the main window (see Figure 3–1), which contains two panels. The navigator or tree panel appears on the left side of the main window and the property sheet panel appears on the right side. When you select an item in the navigator panel, the property sheet panel displays the properties of the selected item.

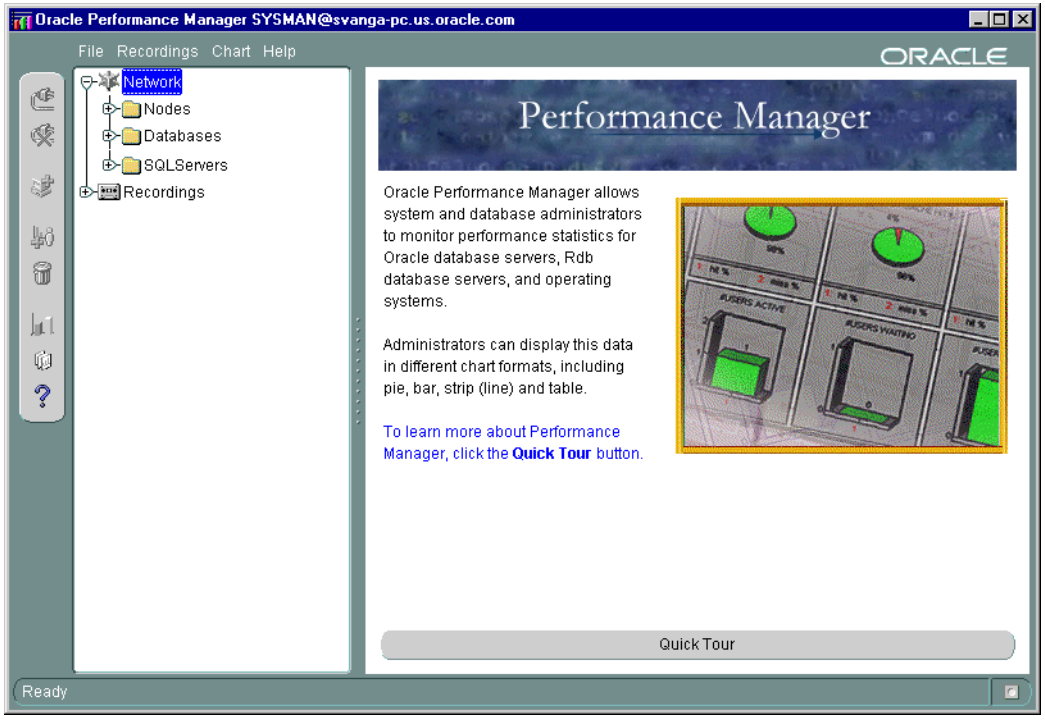
When you first start Oracle Performance Manager, the navigator panel displays the service type folders and the property sheet panel displays the Performance Manager welcome screen.

A vertical toolbar appears on the left side of the window and a menu bar appears across the top of the window. For more information about the buttons in the toolbar and the commands available from the Performance Manager menus, choose **Window** from the Performance Manager **Help** menu.

In the navigator panel, the **Network** folder contains a folder for each service type. For example, when you have the Diagnostics Pack installed, the service types are Nodes, Databases, and SQL Servers.

In addition to the **Network** folders, the **Recordings** folder contains any recordings of chart data that exist. For more information about the **Recordings** folder, see "Recording Chart Data" on page 3-19.

Figure 3–1 Oracle Performance Manager Main Window



Connecting to a Discovered Service in Repository Mode

If you are working in repository mode, expand a service type folder to view the corresponding services previously discovered through the Oracle Enterprise Manager console.

To connect to a discovered service, select the service in the navigator tree and click **Connect to Service** in the Performance Manager toolbar, or click the plus sign (+) next to the service in the navigator. You may be prompted with a Logon dialog box if credentials are required to connect to the service. For information on the fields in the Logon dialog box, click **Help**.

Manually Connecting to a Service

If you are working in standalone mode, the service type folders are empty and you must manually add each service you want to monitor to the navigator tree.

If you are working in repository mode, you can manually add additional services that have not been discovered using the Enterprise Manager console.

> **Note:** When you add a service manually to the navigator tree using the **Add New Service** button, you are adding that service for the current session of Performance Manager only. You can begin collecting data for the service, but if you exit and restart Performance Manager, you will have to add the service again.

To add and connect to a service manually:

1. Select the folder for the service type you are adding.
2. Click **Add New Service** in the Performance Manager toolbar panel.

   Performance Manager displays a Logon dialog box.
3. Fill in the fields in the Logon dialog box and click **OK**.

   For information on the fields in the Logon dialog box, click **Help**.

After you connect to a service, Performance Manager displays the classes of performance information you can monitor for the selected service. Expand a class to see the individual performance data charts available for the selected class.

Viewing Performance Manager Charts

You can display a chart using one of the following methods:

- Select the chart in the navigator, and click the **Show Chart** button on the toolbar.
- Select the chart in the navigator, and choose **Show Chart** from the **Chart** menu.

- Select the chart in the navigator, which displays the chart's property sheet. To start the chart with the selected property sheet options, click **Show Chart** on the property sheet.
- Select a chart class, then select the chart in the **Predefined Displays** list and click **Show Chart**.
- Select the chart in the navigator and click the right mouse button. From the context menu, click **Show Chart**.

Performance Manager displays each chart or chart group in its own window, which is separate from the main window. This window is called the Chart Display

window. For more information, see "Using the Chart Display Window" on page 3-11.

Monitoring Multiple Services Concurrently

You can view charts from multiple services concurrently by connecting to the services from the navigator tree and displaying the chart.

Viewing Chart Groups

Some types of data include a chart group, which is two or more charts that are displayed within a single Chart Display window. You display a chart group the same way as you view any other chart (see "Viewing Performance Manager Charts" on page 3-8).

You can display a special chart group that contains the Overview Charts for a selected service:

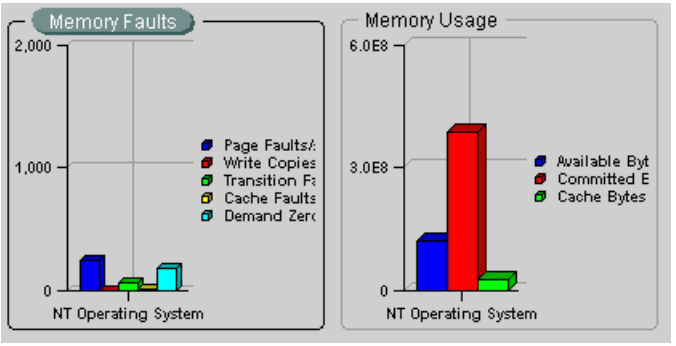
1. Select a service in the Performance Manager navigator.

2. Click the **Show Overview Charts** button in the toolbar panel.

   Performance Manager displays a chart group containing some key statistics most likely of interest to database administrators monitoring the selected service.

Charts in a chart group are displayed within a single window. You perform actions, such as setting options and displaying drill-down charts, on one chart at a time within a chart group. The currently selected chart within a chart group window is identified by a highlighted title. For example, the **Memory Faults** chart is the currently selected chart in Figure 3–2.

Figure 3–2 Identifying the Selected Chart In a Chart Group



Besides viewing the Overview Charts for a selected service, you can also view a chart group provided with a data class. For example, you can select the **Memory** data class for a Windows NT node and display the **Memory at a Glance** chart group. Not all data classes provide chart groups, however.

Selecting Collected Data and Data Sources

For some charts, the chart's property sheet allows you to select the collected data to display and the data sources for which the collected data should be displayed.

The collected data are the statistics that can be monitored and displayed in the chart. The data sources are the items from which Oracle Performance Manager is able to collect the data. For example, the data source of disk statistics may be disk1 or disk2, and the data source of file statistics may be file1 or file2.

By clicking in the column next to the name of the collected data or the name of the data source in the chart property sheet, you can specify whether or not to monitor that collected data or data source and display it in the chart.

When a green check mark appears in the column, Oracle Performance Manager will display that collected data or data source in the chart. If you click in the column again, the check mark goes away, which indicates that Oracle Performance Manager will not display that collected data or data source. You can toggle back and forth between monitoring and not monitoring all the collected data or data sources by clicking in the column head. If the check mark is gray, it means Oracle Performance Manager will display the collected data or data source, and you are unable to change the selection of this data.

If the chart definition contains default data source or collected data attributes, then those items are preselected in the chart's property sheet.

Using the Chart Display Window

When you display a chart, Oracle Performance Manager opens the Chart Display window. The window contains its own toolbar and menu bar. For information about the Chart Display window, choose **Window** from the Chart Display window **Help** menu.

Selecting a Chart Format

To change the format of the chart, click one of the chart format buttons in the Chart Display window toolbar.

For example, suppose a chart initially displays as a bar chart. To see the data as a pie chart, click the **Pie Chart** button in the toolbar. You can also display a chart in one of the additional formats:

- Bar
- Strip (or line chart)
- Table
- Hierarchical

For more information about the buttons on the Chart Display window toolbar, see the Oracle Performance Manager online help.

Note: Depending upon the type of chart you are displaying, one or more of the chart formats may not be available.

Setting the Chart Refresh Rate

By default, most charts are set to refresh every five seconds. That means that Performance Manager collects the data every five seconds and then immediately refreshes (or redisplays) the chart using the new data.

To change the refresh rate for the current chart:

1. Click the **Set Refresh** rate button in the Chart Display window toolbar.

   Performance Manager displays the Set Refresh Rate dialog box.

2. Use the fields on the dialog box to set a new refresh rate for the current chart.
3. Click **OK** to close the dialog box and display the chart using the new refresh rate.

Pausing and Restarting a Chart

By default, Performance Manager continues to collect performance data and update the current chart using the current refresh rate.

To pause a chart so it stops redisplaying based on the refresh rate, click the **Pause Chart** button in the Chart Display toolbar.

To restart the chart, click the **Resume Chart** button in the Chart Display toolbar.

Selecting Chart Options

After you display a chart in the Chart Display window, you can still modify the collected data and data sources for the chart. For more information about selecting the collected data and data sources before you display a chart, see "Selecting Collected Data and Data Sources" on page 3-10.

To set the chart options:

1. Click the **Set Options** button in the Chart Display toolbar.

   Performance Manager displays the Set Options dialog box.

2. Use the dialog box to add or remove data items from the chart.

   For more information on selecting data sources and selecting collected data, click **Help**.

Oracle Performance Manager also allows you to select filters to limit the data collected for a chart. However, not all charts provide filters.

For charts that allow you to filter the chart data, you can select a limit to the number of data sources that are displayed. For example, if the data sources are the processes on an operating system, you can modify the chart to display only 10 processes, rather than all the processes.

For example, to select a limit to the number of data sources that are displayed for the **Pagefaults Per Session** chart, do the following:

> **Note:** The **Pagefaults Per Session** chart in the **Process** class is available only when you are monitoring a Windows NT node. For a UNIX node, select the **Process Info** chart in the **Process Data** class as an example of a chart you can filter.

1. Highlight the **Pagefaults Per Session** chart in the navigator tree of the Performance Manager main window.

   You can locate the chart by traversing the tree as follows: **Network**=>**Nodes**=><Name of Node>=>**Process**=>**Pagefaults Per Process**.

2. Click the **Options** tab in the Property Sheet panel.

3. Enter **10** in the **Maximum Number of Data Sources to Display** field.

   The 10 data sources will be sorted by Page Faults per second, as shown in the **Sort the data by** drop-down list.

> **Note:** The number of options available on the Options tab will vary, depending upon the class of data you are modifying. Some data classes offer more or fewer options than the Pagefaults Per Process class discussed here.

4. Click **Show Chart**.

   Performance Manager displays the chart, but includes a maximum of 10 data sources.

You can later change this setting after the chart is displayed:

1. In the Chart Display window for the Pagefaults Per Process chart, click the **Set Options** button in the toolbar.

2. In the **Options** tab, set the **Maximum Number of Data Sources to Display** back to 7.

3. Click **OK**.

These options can be useful for finding the top resource users. In the case where processes are the data sources and the data collected is memory usage and CPU usage, you can ask the Agent data gathering service to limit the number of data

sources to 10, and to sort by memory usage. Your chart would then show the 10 processes using the most memory. If you were to sort by CPU usage, you would get the 10 processes using the most CPU instead.

Some charts can also give different views of the data. If that feature is available, you will also be able to select a view of the data. An example of a data view may be the current rate per second or current rate per transaction for each piece of data collected in the chart.

Chart-Specific Commands

Some types of charts have their own unique commands. For example, you can select the **Kill Session** command when you are viewing the Lock Manager charts for Oracle database services.

To see if a chart-specific command is available for a chart:

- Click the right mouse button on an item name in the chart legend or a data item in the chart and click **Tools** on the context menu.
- Click the **Tools** menu on the Chart Display window menu bar.

Drilling Down from One Chart to Another

Some charts have drill-down charts associated with them. To see if any drill-down charts are available for the current chart, do one of the following:

- Click the right mouse button on an item name in the chart legend or a data item in the chart and click **Drilldown** on the context menu.
- Click the **Drilldown** menu on the Chart Display window menu bar.

In either case, Performance Manager displays any drill-down charts associated with that chart item as menu options on the **Drilldown** menu. If a list of drill-down chart names is not displayed, it means no drill-down charts are associated with that item.

Note: If you have trouble selecting a particular data item when you try to drill down for related data, pause the chart or reduce the refresh rate. For more information, see "Pausing and Restarting a Chart" on page 3-12.

Drilling Down to an Historical Chart

From a Performance Manager real-time chart, you can drill down to see an historical view of the same data collected over a longer period of time, for example, days, months, and years. This view of historical data, provided by the Oracle Capacity Planner software, allows you to analyze long-term trends in the data.

Historical data collection must be enabled to view the historical data for a chart. If you drill down to a historical chart for which historical data collection has not been enabled, Performance Manager allows you to start historical data collection. You can also start historical data collection by starting Oracle Capacity Planner. From the Capacity Planner window, you can also control other aspects of your historical data collection, such as how often data is collected and loaded into the historical database.

Note: User-Defined charts do not support drill-down to historical data.

To drill down to historical data:

1. Display a chart in the Chart Display window.
2. Click the right mouse button on a data item in the chart.
3. From the context menu, click **Drilldown**.
4. From the **Drilldown** submenu, click **Historical Data**.

What happens next depends on whether or not you have begun collecting historial data. Consider three possible scenarios:

- If Performance Manager finds historical data related to the chart, Performance Manager displays the Analysis Chart window.

  Note that if you previously set up your historical database using an alternate database location, Performance Manager displays a database logon box before it displays the Analysis Chart window.

  The Analysis Chart window shows you the historical information about the collected data. From here, you can modify the format of the chart or perform a trend analysis on the historical data. For more information, select an option from the **Help** menu in the Analysis Chart window or see the chapter about Capacity Planner in this manual.

- If Performance Manager cannot find historical data related to the chart, the following message appears:

  ```
  The historical chart cannot be displayed because there is no history
  available for the data sources or data items.

  Would you like to begin collecting the history of these data sources and
  data items now so that you can view the chart in the future?
  ```

  Click **Yes** to begin collecting historical data automatically or click **Help** to learn how to collect historical data using the Oracle Capacity Planner software.

  When you click **Yes**, Performance Manager displays another message explaining that historical data is now being gathered, but is not yet available. Click **Continue** to return to the Display Chart window.

- If no historical data has ever been saved to the historical database, Performance Manager displays a message box that includes the following errors:

  ```
  Error connecting to the historical database.
  The specified schema is not a Capacity Planner database.
  ```

  Click **Continue**. Performance Manager displays the message described in the previous bullet item.

  Click **Yes** to begin collecting historical data or click **Help** for more information about how historical data is collected. When you click **Yes**, Performance Manager displays another message explaining that historical data is now being gathered, but is not yet available.

  Click **Continue** to return to the Display Chart window.

Displaying Alternate Chart Orientations

You can display bar charts with either a horizontal or vertical orientation.

To change the orientation of the bar chart, click the button associated with the orientation that you want.

Sorting Data in a Table Chart

When a chart is displaying data in table format, the sort order of the column data can be changed by clicking the column headers in the table. The first click on a column header sorts the data in ascending order, and a second click sorts the data in descending order.

Changing the sort order in the chart table does not affect the data's sort order on the server. To change the sort order of the data on the server, use the **Options** page. For more information, see "Selecting Chart Options" on page 3-12.

Saving Your Changes to a Chart

If you are working in repository mode and you make modifications to a chart (for example, if you change the format, orientation, or sorting order of a chart), you can save your changes in one of two ways:

- Click **Save** on the Chart Display **File** menu.

  Performance Manager saves your changes using the same name as the original chart. When you display the chart again, your changes will appear in the Chart Display window.

  To later remove your changes and display the predefined chart using its original settings, click **Remove Chart** on the **Chart** menu.

  Note: You can remove the changes you saved to a predefined chart, but you cannot delete the original predefined chart.

- Click **Save As** on the Chart Display **File** menu.

  Performance Manager displays the Save Custom Chart dialog box. Use this dialog box to enter a name for the custom chart and to indicate whether you want to create a similar chart for other services of the same type. For information about the fields on the Save Custom Chart dialog box, click **Help**.

  Click **OK** in the Save Custom Chart dialog box to save the custom chart using the name you provided. Performance Manager saves the chart in the Custom Charts folder for the selected service.

  To delete a custom chart from the Custom Charts folder, select the chart and click **Remove Chart** on the **Chart** menu.

  Note: You can delete custom charts, but you cannot delete predefined charts.

Obtaining Help for a Class, Chart, or Data Item

To get help on a class or chart:

1. Select the class or chart.
2. From the **Help** menu, choose **Selected Item**.

   If help is available for the selected chart, Performance Manager displays a message box or a help topic window describing the chart data.

To get help on a data item:

1. Select a chart.
2. In the Property Sheet panel, hover over a data item or select a data item and choose **Selected Item** from the **Help** menu.

   If help is available for the selected item, Performance Manager displays a pop-up balloon or a message box describing the data item.

Note: Help is not available for all objects.

Printing a Chart

To print a chart, click the **Print Screen** button on the Display Chart window toolbar.

The size of the chart on the screen determines the size of the chart in the chart printout. For example, when you print a chart that is 5 inches high and 7 inches wide on your screen, the chart will be 5 inches high and 7 inches wide in the printout.

Generating a Report for a Chart

To generate an HTML report for a chart:

1. Click the **Report Chart** button while the chart is displayed.

   Performance Manager displays a dialog box that tells you the name of the HTML file used to display the report and where the file will be saved.
2. Make a note of the file name and location.
3. Click **Yes** to preview the chart in your Web browser, or click **No** if you do not want to preview the report.

You can view the report later using a Web browser, or—if you have experience with Web publishing tools and techniques—you can share the report with your co-workers by publishing the HTML file and its associated graphic files on a Web server.

Recording Chart Data

Oracle Performance Manager allows you to record the data being monitored in a chart. After a recording is complete, you can play it back. During playback, you can pause and fast forward the recording, if you wish.

Chart recordings are saved under the **Recordings** folder and on the Recordings tab of the right pane of the Performance Manager main window when a recorded chart is selected. The Recordings tab is added to a chart's property sheet after you record the chart.

You can manage chart recordings in the **Recordings** folder without connecting to the service from which a chart was recorded. If you are already connected to a service, you can see a chart's recordings by selecting the **Recordings** tab in that chart's Detail property sheet.

Starting a Recording

You start a recording by performing one of the following steps:

- With a chart highlighted in the navigator tree of the Performance Manager main window, click **Record Chart** on the chart's property sheet.

- With a chart displayed in the Chart Display window, click **Start Recording** in the Chart Display window toolbar.
- With a chart displayed in the Chart Display window, click **File**=>**Recording**=>**Start Recording**.

Oracle Performance Manager displays the Recording Parameters dialog box. For more information on the fields in the Recording Parameters dialog box, click **Help**.

Stopping a Recording

To stop recording a chart, do one of the following:

- To stop recording the current chart, click the **Stop Recording** button in the Chart Display window toolbar.

- To stop all recordings at one time, choose **Stop All Recordings** from the Performance Manager main window **Recordings** menu.
- To stop a recording from the **Recordings** folder, expand the **Recordings** folder in the Performance Manager main window and navigate to the chart. Select the chart and click **Stop Recording** on the chart's property sheet.
- With a chart displayed in the Chart Display window, click **File**=>**Recording**=>**Stop Recording**.

Playing a Recording

Note: Before you can play a recording, you must stop the recording. For more information, see "Stopping a Recording" on page 3-19.

To play a recording, do the following in the Performance Manager main window:

1. Navigate to the **Recordings** folder.
2. Select the chart you want to play in the navigator tree.
3. In the corresponding property sheet, select the recording from the list of recordings for the selected chart, and click **Play Recording** (or on the **Recordings** menu click **Play Recording**).

4. In the Chart Display window toolbar, click **Start Playback**.

   The **Stop Playback**, **Pause Playback**, and **Fast Forward** buttons become available in the playback window (Figure 3–3). When the playback is complete, the **Reset Playback** button becomes available.

Figure 3–3 Chart Playback Buttons in the Chart Display Window

Removing a Recording

To remove a recording:

1. Navigate to the **Recordings** folder or the Recordings tab on the chart's Detail page.
2. Select the recording you want to remove.
3. From the **Recordings** menu, click **Remove Recording**.

Creating a User-Defined Chart

In addition to predefined charts, you can define your own charts for database services based on user-defined scripts. By defining your own charts, you can take a snapshot of data at any given interval. After successful evaluation of the script, you can define the operations to be performed on the data.

Note: The Performance Manager online help includes an example of creating a user-defined chart. Search for **Example** in the online help index.

To define a new chart:

1. Expand the **Databases** folder in the navigator tree.
2. Expand the database for which you want to define a user-defined chart.
3. Select the **User-Defined Charts** folder.

4. Click **Add User-Defined Chart** in the toolbar.

   The resulting New Chart property sheet includes the SQL Script Command page and the Data Items page.
5. On the SQL Script Command page, enter a SQL script that will be used to gather statistics for the chart you are defining.
6. Click **Evaluate**.

   Performance Manager evaluates the SQL script. If the script is valid, Performance Manager displays the Data Items page.
7. Use the Data Items page to customize how the resulting data from the script entered on the SQL Script Command page will be displayed in chart form.

You can remove columns and change the names of the columns. You can also define new columns with data calculated from existing columns. For more information, see "Modifying the Data Items for a User-Defined Chart" on page 3-22.

Modifying an Existing User-Defined Chart

To modify an existing chart you perform the same steps as outlined in "Creating a User-Defined Chart" on page 3-21. The only exception is that the chart label is already defined.

Note: To add or modify data items to user-defined charts, you must use the Agent data gathering service release 8.1.6. or later.

To modify an existing chart:

1. Expand the **Databases** folder in the navigator tree.
2. Expand the database for which you want to modify a user-defined chart.
3. Select the **User-Defined Charts** folder.
4. Click the name of the chart you want to modify.

   On the SQL Script Command page, you can modify the script used to gather statistics for the chart. On the Data Items page, you can remove, add, or modify the data items associated with the chart. For more information, see "Modifying the Data Items for a User-Defined Chart" on page 3-22.

Modifying the Data Items for a User-Defined Chart

When you select a user-defined chart and then display the Data Items page, you can modify the data items for the chart as follows:

- Select a data item and click **Set Data Source**.

A key icon appears next to the data item you designate as the data source. The data source is the object about which Performance Manager is able to collect the data. You must designate one of the data items as the source for the user-defined chart. To designate a data item as the data source, highlight the data item you want as the data source and click this button. By default, the first data item is set as the data source.

- Select a data item and click **Remove**.

  Performance Manager removes the data item from the Data Items page.

- Select a data item and click **Modify** or click **Add** to add a new data item to the user-defined chart.

  Note: To add or modify data items to user-defined charts, you must use the Agent data gathering service release 8.1.6 or later.

Performance Manager displays the Add/Modify Chart Columns dialog box. For more information about this dialog, click **Help**. The Oracle Performance Manager Help provides details on the functions and operations available. The Help also provides an example of creating a user-defined chart.

Copying a User-Defined Chart

You can copy user-defined charts from one service to another or within the same serve by using the Create Like option. To copy a user-defined chart:

1. In the navigator tree, select the user-defined chart you want to copy.
2. From the **Charts** menu, click **User-Defined Charts**=>**Create Like**.

   Performance Manager displays the Create Like User-Defined Chart dialog box.
3. Type a label for the new chart in the **Chart Label** field.

   Note that chart labels must be unique within a service.
4. In the **Service** drop-down list, select the target service for the new chart.
5. Click **OK**.

In the Create Like User-Defined Chart dialog box, you can choose the SQL Script Command or Data Items page to view the attributes of the chart you are copying. However, you cannot change the values of the attributes. Once the chart has been copied, you can select the new chart in the navigator tree and make any desired adjustments.

Converting Old Performance Manager User-Defined Charts

If you created user-defined charts using Oracle Performance release 1.5.0 or earlier (the Windows versions of Oracle Performance Manager), you can convert these

user-defined charts so that they can be used with Oracle Performance Manager release 2.1. See "Converting User-Defined Charts From Previous Releases of Performance Manager" on page 2-3 for information on how to perform the conversion.

4

Introduction to Oracle Capacity Planner

Oracle Capacity Planner is an Oracle Enterprise Manager application that allows you to collect different types of performance statistics and record that data in an historical database. You can then use Oracle Capacity Planner to analyze the historical data to plan future capacity.

Oracle Capacity Planner uses the Agent data gathering service (also called the Oracle Data Gatherer) to collect its historical data. To collect most types of data, you must install the Oracle Data Gatherer on the system (or systems) where you want to collect data. See the *Oracle Enterprise Manager Intelligent Agent User's Guide* for information on managing the Oracle Data Gatherer.

If you are using Oracle Capacity Planner to collect database data, the Oracle Data Gatherer can be installed on another system. See "Accessing Historical Data Through an Intermediate Host" on page 4-3 for more information on accessing data through an Oracle Data Gatherer on another system.

The types of performance data that Oracle Capacity Planner can collect on a system depend on the products that are installed. Table 4–1 shows the types of data that Oracle Capacity Planner can collect when various Oracle products are installed.

Table 4–1 Data That Capacity Planner Can Collect When Various Oracle Products are Installed

Oracle Product	Oracle Concurrent Manager Data	Oracle Database Data	Node (Operating System) Data	SAP R/3 System Data	Microsoft SQL Server Data
Diagnostics Pack		Yes	Yes		Yes

Table 4–1 Data That Capacity Planner Can Collect When Various Oracle Products are Installed (Cont.)

Oracle Product	Oracle Concurrent Manager Data	Oracle Database Data	Node (Operating System) Data	SAP R/3 System Data	Microsoft SQL Server Data
Management Pack for Oracle Applications	Yes		Yes		
Management Pack for SAP R/3			Yes	Yes	

If additional products from Oracle or other vendors are installed, Oracle Capacity Planner may be able to collect additional types of data not shown in Table 4–1.

How Capacity Planner Collects and Stores Performance Data

The following list identifies how Capacity Planner uses the Agent data gathering service to collect performance data:

- The Agent data gathering service collects the data at an interval you specify from within Capacity Planner.

  The Agent data gathering service stores the data in an efficient binary form.

- At a specified loader interval, the data is read from this binary file and stored in a readable form in the Oracle Capacity Planner historical database.

  The Agent data gathering service also aggregates the data by time, creating summary statistics for the data that is collected at different time intervals (hour, day, week, and so on).

- Over time, the Agent data gathering service purges the data in the historical database, keeping only as much data as you specify.

  Purging the historical data allows you to have access to a current view of recent performance data for your environment and to manage the amount of stored data to avoid wasting disk space.

Accessing Historical Data Through an Intermediate Host

Oracle Performance Manager relies on the Oracle Data Gatherer to monitor a service (such as a node, a database, or a concurrent manager). Usually, the collection of the data is done on the system where the service is located. However, for some types of services Oracle Performance Manager can also monitor data on a system by using the Oracle Data Gatherer on another system (an intermediate host).

This intermediate host could be the client system where the Oracle Performance Manager is running, or any other system on the network on which the Data Gatherer is available.

There are three ways to set the location of the Oracle Data Gather. The method you use depends on how you connect to the service. Consider the following scenarios:

- You connect to a database that has been discovered using the Enterprise Manager console and you are running Oracle Performance Manager while connected to the Oracle Management Server.

  Performance Manager attempts to connect to the Data Gatherer on the host where the service is located. If this connection fails or cannot be attempted because no preferred credentials have been set for the database, you will be prompted with a dialog box containing logon credentials for the database. This dialog box also contains a field to specify the location of the Oracle Data Gatherer that will be used to collect performance data.

- You click the **Add Service** button in the toolbar panel and manually add the service to the navigator.

  Performance Manager displays a Logon dialog box. This dialog box also contains a field to specify the location of the Data Gatherer service that will be used to collect performance data.

- You select a service in the Performance Manager navigator tree and choose **Set Data Gatherer Location** from the **File** menu.

  Performance Manager displays the Database Logon dialog box for the selected service. This dialog box also contains a field to specify the location of the Data Gatherer that will be used to collect performance data.

Note: The database name (or service) shown in the Database Logon dialog box must be a service name which can be resolved on the system where the Agent data gathering service is located. A **tnsnames** entry or name server entry must be available for that service on the system specified in the **Data Gatherer Location** field.

Usage Scenarios for Oracle Capacity Planner

Oracle Capacity Planner can perform tasks such as the following:

- Track the increase in the amount of used space on a disk over time and estimate when 95 percent of the space on the disk will be consumed
- Track the increase in the total I/O rate for a host over time, then estimate what the total I/O rate for the host will be at some future point in time

Starting Oracle Capacity Planner

You can start Oracle Capacity Planner in any of the following ways:

- On the Oracle Enterprise Manager console, click the **Diagnostics Pack** drawer, then click the **Capacity Planner** icon.
- On the Oracle Enterprise Manager console, select a Database service, click the right mouse button, point at **Related Tools** in the context menu, then click **Capacity Planner**.
- On the **Tools** menu of the Oracle Enterprise Manager console, point to **Diagnostics Pack**, then click **Oracle Capacity Planner**.
- From the **Start** menu, point to **Programs**, then to the Oracle Home where Oracle Enterprise Manager is installed, then to **Diagnostics Pack**, then click **Capacity Planner**, which displays the Capacity Planner Login dialog box.

  Enter the appropriate connect information for your Oracle Enterprise Manager repository, then click **OK**.

Using Help

When you are using Oracle Capacity Planner, you can display context-sensitive help topics. You can also display conceptual topics and task topics.

To obtain context-sensitive help for an Oracle Capacity Planner panel, window, dialog box, or property page, select an item in the panel, window, dialog box, or property page, then on the **Help** menu, choose **Window**.

To obtain context-sensitive help for a class, chart, or data item on a property page, select the class, chart, or data item on the property page, then take one of the following actions:

- Press F1.
- Click the **Help** icon on the toolbar.
- On the **Help** menu, choose **Selected Item**.

If a help topic is not available for a class, chart, or data item, a No Help Available message is displayed when you request help.

To view the Contents page for the Oracle Capacity Planner help, on the **Help** menu, choose **Contents**. On the Contents page, you can:

- View a list of the conceptual help topics by double-clicking the **Conceptual Topics** book on the help system's Contents page.

  Conceptual topics provide conceptual or overview information of which you should be aware before performing a particular Oracle Capacity Planner task. Conceptual topics do not contain step-by-step information for performing a task.

- View a list of the task help topics by double-clicking the **Task Topics** book on the help system's Contents page.

  Task topics provide step-by-step instructions for accomplishing an Oracle Capacity Planner task. They do not provide conceptual information that you should be aware of before performing the task.

You can access a glossary of Oracle Capacity Planner terms by double-clicking the **Oracle Capacity Planner Glossary** book on the Contents page.

Using Oracle Capacity Planner

You may find it helpful to think of using Oracle Capacity Planner as a two-step process:

1. Setting up the collection of historical data

   See "Setting Up a Collection" on page 4-6 for a detailed description of how to collect historical data.

2. Analyzing the collected data

   See "Analyzing Collected Data" on page 4-26 for a detailed description of how to analyze collected data.

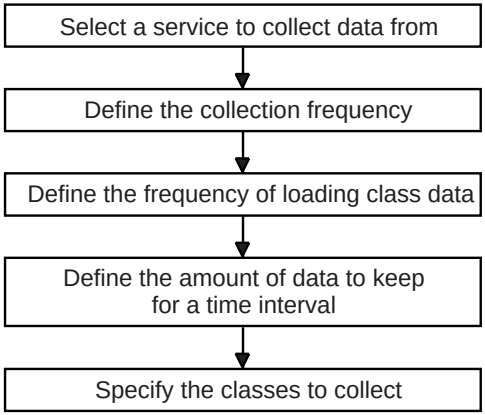
Setting Up a Collection

To set up a historical data collection with Oracle Capacity Planner, you:

1. Select (and connect to) the services for which you want to collect historical data.
2. Define the frequency of collection samples.
3. Define the location of the historical database and the frequency at which collected data should be loaded in the historical database.
4. Define the amount of data to be kept in the Oracle Capacity Planner database at each time interval.
5. Select the classes of data to collect, based on what is available in the Oracle Capacity Planner navigator.

Figure 4–1 shows the steps to follow when setting up a collection. For more information on each step, see the following sections.

Figure 4–1 Steps for Setting Up a Collection



Selecting Services

You can select one or more services from which you want to collect historical performance statistics. For example, when you have the Oracle Diagnostics Pack installed, the service types are usually Nodes and Databases.

The procedure for selecting a service depends upon whether or not the server resides on a node that is known to the Enterprise Manager main console. A node is known to the console when it has been *discovered* with the Enterprise Manager console **Discover Nodes** command.

Selecting Services on Discovered Nodes

To select a service on a node that has been discovered by the Enterprise Manager console:

1. Click the **Collection** tab of the Oracle Capacity Planner navigator panel, which displays the Oracle Capacity Planner collection view.
2. Double-click one of the service type folders.

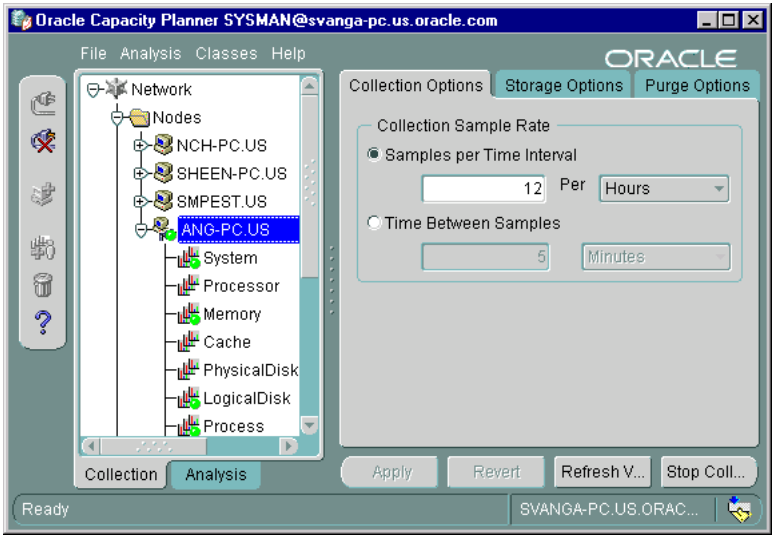
   The Oracle Capacity Planner navigator tree displays the services of the selected type that are known to the Oracle Enterprise Manager main console.

3. Select the service from which you want to collect performance data and click the **Connect** button in the Capacity Planner toolbar, or click the plus sign (+) next to the service in the navigator.

When you connect to a service in the Oracle Capacity Planner navigator panel, you may be prompted for credentials if they are required to connect to that service. If a logon dialog box appears, click **Help** for information on the individual fields on the dialog box.

After you connect to a service, the available collection classes for that service appear in the navigator and Oracle Capacity Planner displays the Service property sheet for the selected service. For example, Figure 4–2 shows the classes that appear when you connect to a Windows NT node service.

Figure 4–2 Selecting a Service in the Oracle Capacity Planner Window



Selecting Services on Undiscovered Nodes

If you double-click a service type and the service you are looking for does not appear in the service type folder, you can do one of the following:

- Use the Enterprise Manager console to discover the node where the service resides.

  This is the preferred method for adding a service to the Capacity Planner navigator panel. When you discover a node with the Enterprise Manager console, Capacity Planner saves key information about the service, such as the preferred credentials, location of the Agent data gathering service, and historical database information.

- Add the service manually to the Capacity Planner navigator panel.

  When you add a service manually, the service is added for the current session of Capacity Planner only. You can start gathering data for the service, but the next time you start Capacity Planner, the service will not appear in the navigator panel. To modify collection settings or analyze data collected for the service, you will have to add the service again the next time you start Capacity Planner.

To add a service manually:

1. Select the service type for the service you want to add.

2. Click the **Add Service** button in the Capacity Planner toolbar.

   Capacity Planner displays a message to remind you that services you add manually will be available for the current session of Capacity Planner only.

3. Click **Yes**.

   Capacity Planner displays a logon dialog box for the service. Click **Help** if you need information about any of the specific fields on the dialog box.

4. After you fill in the logon credentials, click **OK** to connect to the service.

   After you connect to a service, the available collection classes for that service appear in the navigator and Oracle Capacity Planner displays the Service property sheet for the selected service (Figure 4–2).

About the Service Property Sheet

When you connect to a service, Oracle Capacity Planner displays the Service property sheet, which contains the following property pages:

- Collection Options page
- Storage Options page
- Purge Options page

The collection, load, and purge settings you select on these pages apply to all the classes in the selected service. You cannot change these settings for a selected class; you can change them only for a selected service.

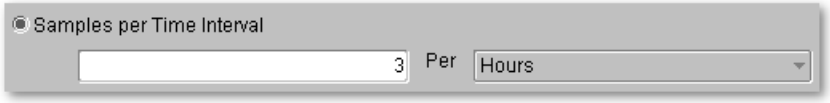
Defining the Frequency of Collection Samples

When you first connect to a service, Capacity Planner displays the Collection Options property page. The values you select on this page determine how often the Agent data gathering service will collect information about the selected service and

save it to disk. On the Collection Options property page, you can set the frequency of collection samples using either of these two methods:

- By selecting the number of data samples to collect for a specified time interval

  For example, to collect information 3 times every hour, select **Samples per Time Interval**, enter 3 in the text field, and choose **Hours** from the drop-down list.



- By specifying the length of time between collection samples

  For example, to collect information every 20 minutes, select **Time Between Samples**, enter 20 in the text field, and choose **Minutes** from the drop-down list.

Note: To configure the settings for collecting data, the Agent data gathering service that will collect data for a service must be running. Normally, the Agent data gathering service is located on the host where the service is located.

To collect data for a database, the Agent data gathering service can be located on a host other than the host where the database is located. See "Accessing Historical Data Through an Intermediate Host" on page 4-3 for more information.

Defining the Location of the Historical Database

After you set the frequency of collection samples, use the **Storage Options** tab to specify the location of the Oracle Capacity Planner historical database. This setting defines where Capacity Planner will save information collected for the selected service. When data is saved in the historical database, you can later analyze the data or use it to generate trend analyses.

By default, collected data is saved in your Oracle Enterprise Manager repository. However, you can specify an alternate database for the historical data if you do not want to store historical data for the selected service in the repository.

To save the data collected for a service in a location other than the Oracle Enterprise Manager repository:

1. From the Storage Options tab, select **Store historical data in alternate location**.
2. Enter the credentials for the database that will store the collection data for the service.

   Enter the appropriate user name, password, and service for the Oracle Capacity Planner historical database. This connection information is used by the Agent data gathering service when it loads the collected data at the specified loader interval. Therefore, the service specified for the Oracle Capacity Planner historical database must be known and accessible from the host where the Agent data gathering service is running.

   For example, if you specify MY_DB.WORLD as the historical database and your Agent data gathering service is running on host MYNODE, then database MY_DB.WORLD must be reachable via Net8 or SQL*Net from host MYNODE (MY_DB.WORLD must exist in the TNSNAMES.ORA file for host MYNODE, for example).

Defining the Frequency of Loading Collection Data

After you define how often data will be collected and where the data will be stored, you can define how often the data will be loaded into the historical database.

You can determine how frequently the Oracle Capacity Planner loader program loads collection data for the service using either of these two methods:

- Using the **Per Time Interval** field to specify the number of loader passes for a specified time interval
- Using the **Time Between** field to specify the length of time between loader passes

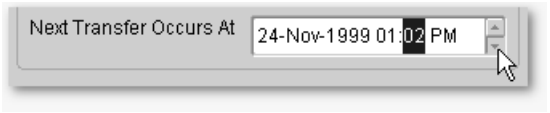
---

Note: Although data can be collected at short time intervals (such as 5, 10, or 15 minutes), collected data is not available to you for analysis until it has been loaded into the Oracle Capacity Planner historical database. Therefore, the load interval you specify controls when the most recently collected data will be visible to you.

The collection frequency you set determines the lowest time interval for which Oracle Capacity Planner can analyze data. For example, if you want to be able to analyze data from hourly intervals, you need to set a collection frequency of at least once an hour.

After you define the frequency of loading collection data, you can set a time for the next loader pass by using the **Next Transfer Occurs at** field:

1. Click the value that you want to change in the date and time box.
2. Click the up or down arrow to select a higher or lower value.



If the current time is later than the time displayed for the **Next Transfer Occurs at** setting, click **Refresh View** to update the display to show the actual time of the next loader pass.

Note that the values that you select on the Storage Options property page apply to all classes of data collected on the selected service.

Understanding When the Loader Runs

The loader runs at the regular time interval specified on the Storage Options property page. The time specified in the **Next Transfer Occurs at** field controls when the loader runs. For example, if you want the loader to run once a day at 2:00 a.m., then set the transfer interval to once per day, and set the next transfer time to the next day at 2:00 a.m. The loader will run for the first time at 2:00 a.m. the next day, and then will run once a day, each day at 2:00 a.m.

You can use the **Next Transfer Occurs** at field to force the loader to run immediately—or as soon as possible. Simply set the next transfer time to a time earlier than the current time.

Understanding How Data Aggregation Works

As it loads data into the database, Oracle Capacity Planner automatically aggregates the data. During aggregation, Oracle Capacity Planner examines the collection samples for a given time interval, calculates an average value for that interval, then uses that average value for calculations at the next larger time interval.

On each loader pass, Oracle Capacity Planner examines the time stamps of all collected data to determine whether or not the threshold for a set interval (hour,

day, week, month, or year) has been crossed since the previous loader pass. If so, Oracle Capacity Planner performs an aggregation of collected data from the next smaller interval to the interval threshold just passed. Table 4–2 shows the larger time interval to which data at each time interval is aggregated.

Table 4–2 Aggregation of Data from Smaller to Larger Time Intervals

Data at This Interval	Aggregates to This Interval
Minutes	Hours
Hours	Days
Days	Weeks
Weeks	Months
Months	Years

An example may help clarify how Oracle Capacity Planner aggregates collection data. Suppose Oracle Capacity Planner is collecting a data sample every 10 minutes to capture the percentage of used space on a particular disk, and the loader runs once an hour. When the Oracle Capacity Planner loader runs and transfers data into the Oracle Capacity Planner database every hour, it examines the values for the percentage of used disk space in the samples collected during the previous hour, then it computes an average percentage for the samples. That average percentage is then used as the average value for the hour.

At the first loader pass after midnight, Oracle Capacity Planner uses the hourly averages since midnight the previous day to compute the day's percentage of used disk space.

At the first loader pass after midnight on Saturday, Oracle Capacity Planner examines the daily averages of used disk space since midnight the previous Saturday and computes an average percentage for the week.

At the first loader pass after midnight on the last calendar day of the month, Oracle Capacity Planner examines the weekly averages since midnight on the last calendar day of the previous month and computes an average percentage for the month.

Finally, at the first loader pass after midnight on December 31, Oracle Capacity Planner examines the last 12 monthly averages to compute a yearly average.

Defining the Amount of Data to Keep

The final step in setting up a service for historical data collection is defining the amount of data to keep.

When you select a service, you can click the **Purge Options** tab to display the Purge Options property page. On this page, you specify the amount of data you want to store in the Oracle Capacity Planner historical database for each type of data (for example, hour data, day data, and week data). You can accept the default Purge Options property page values or choose different values.

The values set on the Purge Options property page apply to all classes of data collected on the selected service.

The values you specify in the data retention table (the **Enter the amount of data you want to keep** table on the Purge Options page) determine how far back in time you can go to analyze data for the different types of data.

Suppose, for example, you always want to be able to view hour data for the previous six weeks. In this case, for the Hours **Data Type** row in the data retention table, you would enter 6 in the **Timespan** column and Weeks in the **To Keep** column. This tells Oracle Capacity Planner that you want to keep hour data for a 6-week period. Oracle Capacity Planner calculates 6 weeks from the current time, and the **Purge Data Older Than** column shows the time span for which hour data is available. Table 4–3 shows the Hours row in the data retention table with this data entered.

Table 4–3 Specifying How Much Data to Keep for a Data Type

Data Type	Timespan	To Keep	Purge Data Older Than
Hours	6	Weeks	(current date minus 6 weeks)

For each row in the data retention table, you can choose the time span to keep data samples for that type of data by entering the desired value in the **Timespan to Keep** column.

Applying and Reverting the Collection Settings for a Service

When you finish setting all the options on the Service property sheet, do the following to apply your collection settings:

1. Click **Apply**.

If you changed the location of the historical database on the Storage Options page, Capacity Planner displays a message box reminding you that any information stored in the previous location will not be available for analysis.

If you don't want to be reminded about this implication the next time you change the database location, select **Don't show this message again**.

2. Click **OK**.

   Capacity Planner displays a message stating that the options were set successfully.

3. Click **Continue** to close the message box and return to the Capacity Planner main window.

When you apply your collection settings, Capacity Planner sends the settings to the Agent data gathering service for the selected service. Note that you cannot select data to be collected until these settings have been applied.

At any time during the process of modifying the settings on the Service property sheet, you may revert the settings to their last saved (applied) state by clicking **Revert**. However, after you apply the settings, they cannot be reverted to their previous settings using the **Revert** button; they must be modified manually.

Selecting the Classes of Data to Collect

After you set the collection and loading options for a service, you can then select the classes of performance data that you want to collect for that service:

1. In collection view, select and connect to the service.

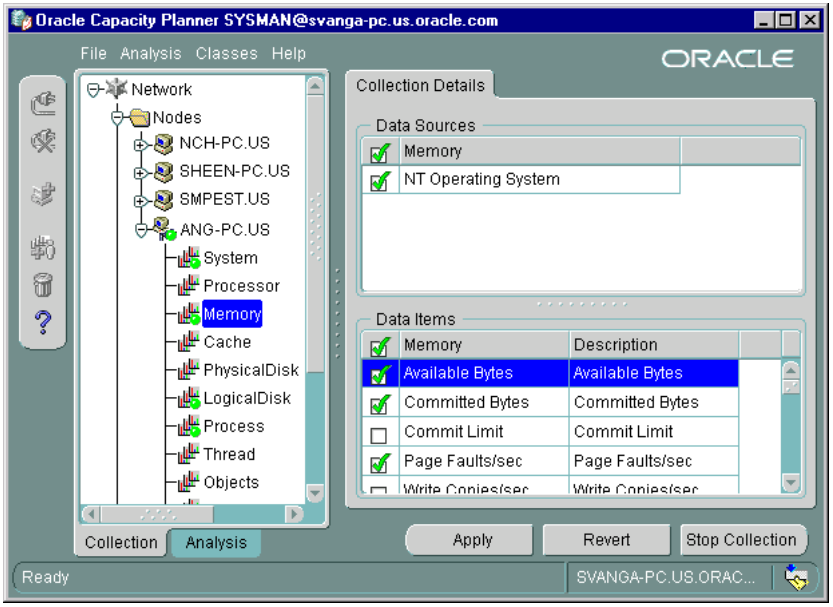
   For more information, see "Selecting Services" on page 4-7.

2. Click the plus sign (+) next to the service to show the classes of data you can collect.

3. Select one of the classes for the service.

Figure 4–3 shows a typical collection class for a node service and the data sources and data items available for the class.

Figure 4–3 Selecting Data Sources and Data Items for a Collection Class



The following list explains classes, data items, and data sources in more detail:

- Classes

  A class is a container for a particular category of data. For example, a node (or operating system) service might include the following classes:

  – System

  – Processor

  – Memory

  When you select a service in the navigator, Capacity Planner displays the names of the classes below the service in the tree view.

  You can collect data from none, some, or all displayed classes for a service. A class may also be a container for other classes, to provide logical groupings of the performance data at higher levels. For example, an I/O class may be a container for other more specific I/O classes.

- Data items

  In a class, each of the related statistics is referred to as a data item. For example, a LogicalDisk class might include the following data items:

  – Percent Free Space

  – Free Megabytes

  – Current Disk Queue Length

  – Disk Writes per Second

  A class can contain one or more data items. When you select a class in the navigator, the names of the data items and data sources associated with the data items are displayed on the Collection Details property page. Select the data items that you are interested in collecting.

  A green check mark appears next to the name of each data item that is selected for collection. You can toggle the state of all data items between collected and not collected by clicking the **Data Items** column header that contains the check mark.

- Data sources

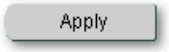
  A data source is a description of where Oracle Capacity Planner can collect the data items for a class. A class can contain one or more data sources. When you select a class in the navigator, the names of the data items and data sources associated with the class are displayed on the Collection Details property page.

  The type of data in the class determines the data sources for the class. For example, in a LogicalDisk class, the data sources might be the individual disks for the operating system (for example, disk1, disk2, and so on). In a CPU Utilization class, the sources are the individual CPUs for the host (for example, cpu1, cpu2, and so on). You can collect data items from one or more data sources.

  When you select a data source, a green check mark appears next to the name of the data source on the Collection Details property page. You can toggle the state of all data sources between collected and not collected by clicking the **Data Sources** column header that contains the check mark.

Starting Data Collection

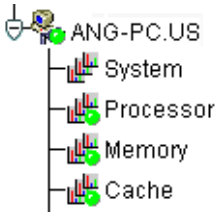
After you select the data items that you want to collect from the available data sources on the Collection Details property page, click **Apply** to confirm your selections, or click **Revert** to cancel the selections.



When you click **Apply** for a class of data, Capacity Planner marks the icon for the selected class and the selected service with a green dot in the Collection view of the navigator panel. A green dot on a collection class icon indicates that one or more data sources are selected for the class. A green dot on a service icon indicates that Capacity Planner is collecting data for one or more classes in the service.

For example, in Figure 4–4, Capacity Planner is actively collecting data for the Processor, Memory, and Cache classes for node ANG-PC. Capacity Planner is not collecting data for the System collection class.

Figure 4–4 Green Dots on a Service or Class Indicate Capacity Planner is Collecting Data



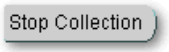
Stopping Data Collection

To stop collecting data for one or more data items or data sources for a collection class, follow these steps:

1. Select the collection class in the navigator.
2. On the Collection Details page, click on the green check marks for those data items or data sources from which you no longer want data collected, which causes the check marks to disappear.
3. Click **Apply**.

To stop collecting data for all data items and data sources for a collection class, follow these steps:

1. Select the collection class in the navigator.



2. On the Collection Details page, click **Stop Collection**.

   Capacity Planner displays a message box stating that collections for the class have stopped.

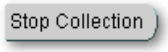
3. Click **OK** to close the message box and return to the Capacity Planner main window.

   When no data sources are selected for collection for a class of data, the icon for that class in the navigator does not display a green dot.

To stop collecting data for all classes in a service:

1. In collection view, select and connect to the service.

   For more information, see "Selecting Services" on page 4-7.



2. Click **Stop Collection** on the Service property sheet.

   Capacity Planner displays a confirmation box.

3. Click **Yes** to stop collecting data for all classes in the service; click **No** to cancel the operation.

Generating a Collection Report

A collection report provides a summary of the collection settings for one or more services. Capacity Planner displays the collection settings in your Web browser. In the browser, you can easily review the settings currently applied to the service or services you selected.

To display a collection report for one or more services:

1. In collection view, on the **File** menu, choose **Collection Report**.

   Capacity Planner displays the first screen of the Collection Report Wizard.

2. Use the Collection Report Wizard to specify the service or services whose collection settings you want to include in the report and to generate and view the report.

   At any time while using the wizard, click **Help** for more information.

When you click **Finish** on the last screen of the Collection Report Wizard, Capacity Planner saves the report to an HTML file and displays the report in your Web browser.

Copying Collection Settings

After you set up collection settings for one servce, you can quickly copy those settings to another similar service.

To copy the collections settings from one service to another:

1. In collection view, select a service whose collection settings you want to copy.
2. On the **File** menu, choose **Copy Collection Settings**.

   Capacity Planner displays the first screen of the Copy Collections Wizard.
3. Use the Copy Collection Settings Wizard to specify the destination services to which you would like to copy the collection settings for the selected service.

   From the wizard, you can also view the progress of the copy operation. At any time while you are using the Copy Collection Settings Wizard, click **Help** for more information.

Note: Any destination service you select must be running an Agent data gathering service of the same type and version as the service from which you are copying your settings.

Creating a User-Defined Class

You can create a user-defined class for any database service. As with other classes, you can collect data from a user-defined class and then analyze that collected data.

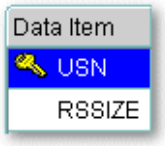
To create a user-defined class:

1. In collection view, select and connect to a database service.

   For more information, see "Selecting Services" on page 4-7.
2. Select the **User-Defined Classes** folder for the database.

3. In the Capacity Planner toolbar, click the **Add User-Defined Class** button.
4. On the SQL Script Command page, specify a name for the user-defined class and a SQL statement that returns the columns of data that you want to include in your user-defined class.
5. Click **Evaluate**.

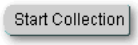
   Oracle Capacity Planner performs a validation check on the SQL statement. When the validation check succeeds, the Collection Details page becomes available.
6. On the Collection Details page, designate a data source for the user-defined class by selecting the data item that contains the column that you want to designate as the data source, then click **Set Data Source**.



A key icon appears in the Data Item column next to the data item that is designated as the data source. For more information, see "Designating a Data Item as the Data Source for a User-Defined Class" on page 4-21.

7. Make any changes that you want to make to the data items by clicking either **Remove**, **Modify**, or **Add**.

   For more information, see "Removing, Modifying, and Adding Data Items in a User-Defined Class" on page 4-22.



8. After you finish defining the user-defined class, click **Start Collection** to start collecting data for the class.

Note: The Oracle Capacity Planner online help includes an example of creating a user-defined class. Search for **Example** in the Capacity Planner online help index.

Designating a Data Item as the Data Source for a User-Defined Class

On the Collection Details page for a user-defined class, you must designate one of the data items as the data source for the class. Each row in the Class Data Items table displays information about a data column that will be returned when the statement on the SQL Script Command page runs. The **Data Item** column displays the name that Oracle Capacity Planner will use for the data item in the user-defined class. The **Derived From** column displays the source or derivation of the Data Item name.

When the Oracle Capacity Planner loader runs and determines that the collected data for the user-defined class needs to be aggregated, the values of the data items are aggregated by the values in the data source, then stored with the data source in the historical database.

For example, the sample user-defined class described in the Capacity Planner online help creates a user-defined class that gathers rollback segment data. In that example, the logical choice for the data source is the rollback segment number. The other data items in the example provide data about each rollback segment. At regular intervals, Capacity Planner aggregates the data items associated with each of the rollback segments (or each rollback segment number) to come up with averages over time.

Removing, Modifying, and Adding Data Items in a User-Defined Class

The Class Data Items table on the Collection Details page displays the data items in the user-defined class. Use the Class Data Items table to remove, modify, and add data items in a user-defined class.

> **Note:** To modify or add data items in a user-defined class, you must be running the Agent data gathering service provided with Oracle Server 8.1.6 or later.

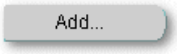
- To remove a data item:

  Select the data item that you want to remove from the class, then click **Remove**. This removes the data item from the Class Data Items table and the user-defined class.

- To modify a data item:

  Select the data item that you want to modify, then click **Modify**, which displays the Add/Modify Collection Details page.

  On the Add/Modify Collection Details page, you can change the label of the data item. You can also specify how the data item is derived. After you finish modifying a data item on the Add/Modify Collection Details page, the information for the data item is updated in the Class Data Items table.



- To add a data item:

  Click **Add**, which displays the Add/Modify Collection Details page. On the Add/Modify Collection Details page, you can specify a label for the data item. You can also specify how the data item is derived. After you finish defining the data item on the Add/Modify Collection Details page, a row for the data item is added to the Class Data Items table.

The Add/Modify Collection Details page allows you to define new data items derived from the definitions of existing data items. You can specify functions to be used on retrieved data items from the script. These function can be performed on numeric data only.

The online help for Oracle Capacity Planner provides details on the functions and operations available.

Modifying User-Defined Classes

Note: You can modify a user-defined class before you start collecting data for the class, but you cannot modify the class after you start collecting data for the class.

If you want to make changes to a user-defined class after starting data collection for the class, create a new user-defined class like the class you want to modify, then modify the new user-defined class. For more information, see "Creating a User-Defined Class Like an Existing User-Defined Class" on page 4-24.

After you make the desired changes to the new user-defined class, you can collect data for the new class and stop data collection for the user-defined class that was created earlier.

To modify a user-defined class before you start collecting data for the class, follow these steps:

1. In collection view, select and connect to a database service.

   For more information, see "Selecting Services" on page 4-7.

2. Select and expand the **User-Defined Classes** folder for the database.

3. Select the user-defined class that you want to modify.

   Capacity Planner displays the SQL Command Script page and the Collection Details page for the class.

4. If you want to modify the name of the class, modify the name in the **Class Label** field on the SQL Script Command page.

5. If you want to change the SQL statement that defines the user-defined class, modify the statement on the SQL Script Command page.

   After you change a SQL statement, click **Evaluate** so that Oracle Capacity Planner can perform a validation check on the statement. When the validation check succeeds, click the Collection Details tab.

6. On the Collection Details page, you can remove, modify, or add rows in the Class Data Items table.

Note: To modify or add data items in a user-defined class, you must be running the Agent data gathering service provided with Oracle Server 8.1.6 or later.

For example, you can click **Remove** to remove a data item, and **Modify** or **Add** to modify or add a data item, respectively.

When you modify or add a data item, you can also specify that functions or operations be performed on data items that have numeric data types. For more information, see the Capacity Planner online help.

Creating a User-Defined Class Like an Existing User-Defined Class

To create a user-defined class like an existing user-defined class, follow these steps:

1. In collection view, select and connect to a database service.

   For more information, see "Selecting Services" on page 4-7.

2. Select and expand the **User-Defined Classes** folder for the database.
3. Select the user-defined class that is similar to the user-defined class that you want to create.
4. On the **Classes** menu, choose **Create Like User-Defined Class**.

   Capacity Planner displays the Create Like User-Defined Class dialog box.

5. Enter a name for the new user-defined class in the **Class Label** field.
6. Select a database service from the **Service** drop-down list.

   Note that you cannot make any other changes to the user-defined class until after you create the new class.

7. Click **OK**.

   Capacity Planner creates the new user-defined class in the User-Defined Classes folder of the service you selected.

   You can now modify the new user-defined class using the procedure in "Modifying User-Defined Classes" on page 4-23.

Starting Data Collection for a User-Defined Class

After you create a new user-defined class, you can start collecting data for the class:

1. In collection view, select and connect to a database service.

   For more information, see "Selecting Services" on page 4-7.

2. Select and expand the **User-Defined Classes** folder for the database.
3. Select the user-defined class for which you want to data.
4. Click **Start Collection**.

   A green dot appears next to user-defined classes when you start collecting data for the class.

Deleting a User-Defined Class

To delete a user-defined class, follow these steps:

1. In collection view, select and connect to a database service.

   For more information, see "Selecting Services" on page 4-7.

2. Select and expand the **User-Defined Classes** folder for the database.
3. Select the user-defined class that you want to delete.
4. In the detail panel, click **Stop Collection**.
5. On the **Classes** menu, choose **Remove User-Defined Class**.

Troubleshooting Collection and Load Problems

During the collection and load processes, errors may be encountered by the Agent data gathering service. You can determine if data is not being collected or loaded by using either of these methods:

- Browsing the analysis data in the Oracle Capacity Planner navigator. See "Browsing the Collected Data" on page 4-27 for more information.
- Accessing the historical database using SQL. See "Understanding the Oracle Capacity Planner Historical Database" on page 4-42 for more information.

If data is not being collected or loaded, you should check the status of the Agent data gathering service on the host from which data is being collected. If the Agent data gathering service is running, check the log file ($ORACLE_HOME\odg\log\alert_dg.log) for errors related to the collection or loading of data.

See the *Oracle Enterprise Manager Intelligent Agent User's Guide* for additional details on the configuration of the Agent data gathering service.

Analyzing Collected Data

After collected data is loaded into the Oracle Capacity Planner historical database, you can perform a variety of analyses on it. To access the analysis features of Oracle Capacity Planner, click the **Analysis** tab (Figure 4–5) at the bottom of the Oracle Capacity Planner navigator tree view. This displays the Oracle Capacity Planner analysis view.

Figure 4–5 Clicking the Analysis Tab

Connecting to an Oracle Capacity Planner Historical Database

When you expand the Historical Database folder at the root of the navigator tree in analysis view, you are prompted for the logon credentials for an Oracle Capacity Planner historical database.

After you connect to a historical database, the folder expands to show the services for which data is available in that historical database.

If you store data in multiple historical databases, you must disconnect from one historical database before you can connect to a different historical database.

To disconnect from the current historical database,

1. Click the Historical Database folder at the root of the navigator tree.

2. Click the **Disconnect** button in the Capacity Planner toolbar panel.

To connect to a different historical database, expand the Historical Database folder again and enter the credentials for the new historical database in the resulting logon dialog box.

Organization of Data in the Navigator

When you expand a service in the navigator under the Historical Database folder, two forms of data are displayed below the service:

- Classes of data

  The classes displayed for the service are the collection classes for which data is available in the Oracle Capacity Planner historical database. You can select and view data from these classes by selecting a class in the navigator, specifying the selection criteria, and then clicking **Show New Chart**. These operations are the first step in creating an analysis.

- Analyses

  Below each service in the navigator is an Analyses folder, which contains predefined and user-defined analyses. Predefined analyses are provided for some services. You can create user-defined analyses by selecting a class of data, displaying an analysis of the data, and then saving the analysis. You can also create a user-defined analysis by using an existing analysis as a template, making modifications to the analysis, and then saving the analysis under a new name.

Browsing the Collected Data

To browse the collected data for a particular service:

1. In analysis view, expand the Historical Database folder.

   You are prompted for the credentials for the historical database that contains the collected data that you want to analyze.

2. Enter your logon credentials and connect to the historical database.

3. On the navigator panel, expand the folder for the service whose collected data you want to analyze.

   The navigator displays the:

   - Names of the classes for which collected data exists in the historical database for the service

- An **Analyses** folder, which contains predefined and user-defined analyses you created for the service
- For database services, a **User-Defined Classes** folder for any data collected from user-defined classes you created in collection view.

Viewing an Analysis for a Class of Data

To view an analysis chart for a selected class of data:

1. On the navigator panel, click the name of a class whose collected data you want to analyze.

   Capacity Planner displays the property page for the class (Figure 4–6).

2. To specify the selection criteria for the class of data you want to analyze, see "Specifying Selection Criteria for a Class of Data" on page 4-29 for instructions.

3. Click **Show New Chart**.

   Capacity Planner displays the Analysis window, which contains two or more pages. The Chart page displays the chart. The remaining pages are Details pages. A separate Details page exists for each class from which one or more data items are included in the analysis.

   For information about the Analysis window, choose **Window** from the Analysis window **Help** menu. For information about modifying the analysis or changing the way the data is displayed, see "Working with Analyses" on page 4-33.

Saving a Class Analysis as a User-Defined Analysis

After you view an analysis for a class of data, you can save it as a user-defined analysis to view later.

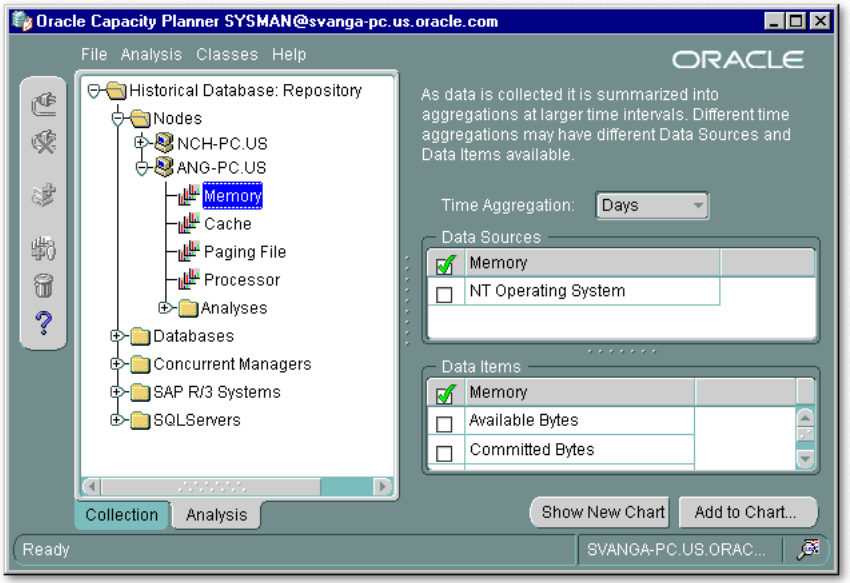
To save the analysis:

1. While viewing the analysis chart, click **Save Chart** on the analysis window toolbar.

2. Click **Yes** in the message box.

   Performance Manager places the new user-defined anlaysis in the **Analyses** folder of the navigator panel.

Figure 4–6 Oracle Capacity Planner Analysis View



Specifying Selection Criteria for a Class of Data

When you select a class in the navigator panel in analysis view, Capacity Planner displays a property sheet that contains options available for selecting data from the Oracle Capacity Planner historical database for that class.

Before you display a chart of the collected data you must:

1. Select the time aggregation level from the **Time Aggregation** drop-down list.

   For example, select **Hours** if you want to analyze hourly data, **Days** if you want to analyze daily data, and so on.

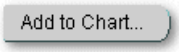
2. In the **Data Sources** section, select one or more data sources whose data items you want to analyze.

3. In the **Data Items** section, select one or more data items you want to analyze.

   A check mark appears next to each data item that is selected for analysis.

After you specify the selection criteria, there are two ways to display the data in an analysis chart:

- To add the selected data to a new analysis chart, click **Show New Chart**, which displays the new chart.



- To add the selected data to an analysis chart that is already displayed in the Analysis window, click **Add to Chart**, then in the Add Data to Analysis dialog box, select the name of the previously-displayed analysis chart. Note that only charts that are currently opened are displayed in the Add Data to Analysis dialog box.

Viewing and Modifying User-Defined and Predefined Analyses

You can view and modify both user-defined and predefined analyses. However, once modifications are made to a predefined analysis, the only way to save those changes is to save the analysis to a new name. This creates a new user-defined analysis and preserves the original predefined analysis.

Note: Not all services provide predefined analyses.

To view an existing analysis, which makes it available for modification, follow these steps:

1. Expand the Analysis folder for the service, which displays the existing predefined and user-defined analyses for the service.

A bar chart icon identifies a predefined analysis. Not all services provide predefined analyses.

A bar chart and person icon identifies a user-defined analysis. You create a user-defined analysis by displaying an analysis for a class of data or displaying a predefined analysis, and then saving the analysis.

Note: You can limit the list of displayed analyses to only user-defined analyses by removing the check mark from the **View Predefined Analyses** option on the **File** menu. When you choose the **View Predefined Analyses** option on the **File** menu again, the check mark preceding the menu option is replaced, and predefined analyses are displayed again.

---

2. Select a predefined analysis or user-defined analysis in the navigator panel.

   The Analysis Summary property page for the selected analysis is displayed in the detail panel. This property page displays the names of the services and classes whose data is used in the analysis.

3. To view the data sources and data items in the analysis, click the plus sign (+) next to the service icons in the **Service/Source** column.

4. To view the chart for the analysis, click **Show**.

   Note that if you are opening a predefined analysis, Capacity Planner displays the Select Data Sources for Analysis dialog box. Predefined analyses are provided for some services. Predefined analyses require you to specify the available data sources for your installation. Select the data sources you want to include in the predefined analysis and click **OK**.

After you have opened an analysis, you can modify the contents and how they are presented. See "Working with Analyses" on page 4-33 for more information.

You can also create a copy of an analysis, rename an analysis, or delete an analysis, as explained in the next three sections.

Creating a Copy of an Analysis

You can create an analysis that is an exact copy of an existing analysis, but the copy has a different name. To do this:

1. In analysis view, locate and expand the service that contains the analysis you want to copy.
2. Expand the **Analyses** folder.
3. Select the analysis you want to copy.
4. On the **Analysis** menu, choose **Create Like**.
5. In the Create Analysis Like dialog box, enter a unique name for the new analysis that you are creating and click **OK**.

   Capacity Planner creates the new copy of the analysis in the **Analyses** folder for the service.

Renaming a User-Defined Analysis

> **Note:** You cannot rename a predefined analysis. This restriction ensures that the original predefined analysis charts are not overwritten.
>
> However, after you make changes to a predefined analysis, you can save the modified version of the predefined analysis as a user-defined chart with a new name. For more information, see "Saving the Results of a Predefined Analysis" on page 4-41.

To rename a user-defined chart:

1. In analysis view, locate and expand the service that contains the analyis you want to rename.
2. Expand the **Analyses** folder.
3. Select the user-defined analysis you want to rename.

A bar-chart-and-person icon identifies user-defined analyses.

4. On the **Analysis** menu, choose **Rename**.
5. In the Rename Analysis dialog box, enter a unique name for the analysis you are renaming and click **OK**.

Deleting a User-Defined Analysis

> **Note:** You cannot delete a predefined analysis.

To delete a user-defined analysis:

1. In analysis view, locate and expand the service that contains the analysis you want to delete.
2. Expand the **Analyses** folder.

A bar-chart-and-person icon identifies the user-defined analyses in the folder.

3. Select the user-defined analysis that you want to delete.
4. On the **Analysis** menu, choose **Delete**.

5. In the message dialog box that prompts you to confirm the deletion, click **OK**.

Working with Analyses

After you have opened an analysis chart, either by selecting and showing an existing analysis or by browsing collected data to create a new analysis, the data included in that analysis is shown in the Oracle Capacity Planner Analysis window.

The Analysis window contains two or more pages. The first page is called the Chart page and it displays the analysis chart. The remaining page or pages are Details pages. A separate Details page exists for each class from which one or more data items are included in the analysis. Each Details page shows the:

- Data sources for the data items plotted in the chart
- Time aggregation level of the plotted data items
- Value of each plotted data item each time the data item was collected during the time range for the analysis

The time range for the collected data appears along the bottom of each chart, with the beginning of the time range at the left and the end of the time range at the right.

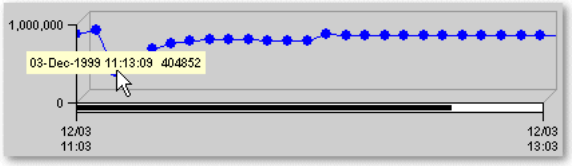
Two types of lines can appear on a chart. Each solid line plots the values of a selected data item during the specified time range. Each dashed line plots the result of a trend analysis to time or trend analysis to value. See "What Is a Trend Analysis?" on page 4-38 for more information about performing a trend analysis.

The rest of this section describes other features of the Analysis window.

Viewing the Value of a Data Point

To view the value for a data point (that is, one plotted point of a data item) on a line in an analysis chart, position the cursor over the data point. If there is a point beneath the cursor, Capacity Planner displays the date, time, and value for that data point displays in a box above the point (see Figure 4–7).

Figure 4–7 Viewing the Value of a Data Point



Selecting a Line

When you view an analysis chart, you can perform certain operations after selecting a line in the chart. Most of the available operations are started by clicking a toolbar button on the analysis window after a chart line is selected.

To select a line, click a data point on the line. When a line is selected, the description of the line is displayed in the status bar at the bottom of the window, and the appropriate toolbar buttons are enabled.

Operations available from the toolbar are also available on a context menu. To display the context menu, click the right mouse button on a data point in a line in the analysis chart.

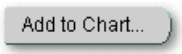
To be sure which line is currently selected (before clicking a toolbar button), read the description of the line in the status bar.

Adding a Class of Data to an Analysis

After you open an analysis, you may want to add data from other classes to the analysis. You can use this feature to compare two different classes of data.

To add a class of data to an analysis that is currently displayed in the Analysis window:

1. Confirm that Capacity Planner is in analysis view.
2. Select from the navigator the class of data that you want to add.
3. Specify the selection criteria (as described in "Specifying Selection Criteria for a Class of Data" on page 4-29) for the class.



4. Click **Add to Chart**.
5. In the Add Data to Analysis dialog box, select the name of the chart to which you want to add the data.

You can add data items from multiple classes to an analysis, and you can select different aggregation levels and date ranges for each class.

However, data items within a single class must share the same aggregation level and date range. If you try to add additional data items from the same class to an existing chart, you will receive a prompt, warning that the chart already contains data from this class. If you click **OK**, Capacity Planner will overwrite the chart with the currently selected data, aggregation levels, and date range.

Modifying Selection Criteria from the Analysis Window

From the analysis window, you can modify the selection criteria associated with any of the classes of data included in the analysis. Modify the selection criteria using one of these methods:

- Select a line in the analysis chart and click the **Select Items** toolbar button.
- Click the right mouse button on a data point in the line and choose **Select Items** from the conext menu.
- Click the **Data** tab in the analysis window (as opposed to the **Chart** tab) and click the **Select Items** toolbar button.

These actions display the Select Items dialog box, which contains the property sheet for the class of data associated with the selected line. This property sheet contains the selection criteria for that class of data, which you can modify and apply to the current analysis. See "Specifying Selection Criteria for a Class of Data" on page 4-29 for more information.

After you modify selection criteria, you can click either **Apply** or **OK** on the Analysis Options dialog box. If you click **Apply**, the changes made to the selection criteria are applied to the analysis and the updated analysis is shown. The Select Items dialog box remains displayed, and you can make other modifications to the same selection criteria, if you want. If you click **OK**, Capacity Planner applies the modifications to the analysis and closes the Select Items dialog box.

Removing a Line from a Chart

You can remove a line from an analysis chart by either of these methods:

- Select the line by clicking the right mouse button, then choose **Remove Line** from the context menu.

- Select the line, then click the **Remove Line** button on the analysis window toolbar.

These actions remove the data for the selected data item from the chart, but not from the historical database.

Excluding a Data Point from a Chart

You can exclude a data point (that is, one plotted point of a data item) from a chart. To exclude a data point, click the right mouse button on the data point and choose **Exclude Point** from the context menu.

Excluding a data point is useful if you know that the value of a particular data point is skewed. For example, if you know that a particular CPU-intensive program was not run at the usual time, you can remove the value for the % Processor Time data item for that time from the chart. The selected data point is removed from the chart, but the data for the data point is not removed from the historical database.

After you exclude one or more data points from a line, a red X appears in the status bar when you select the line.

To later redisplay excluded data points for a selected line, click **View Exclusions** on the Analysis window toolbar, or click the right mouse button on the line and select **View Exclusions** from the context menu.

These actions display the Data Point Exclusions dialog box, which displays the current list of excluded points for the selected line. You can toggle the check mark for each data point in the list. A check mark indicates that the data point is excluded from the analysis. To display an excluded data point in the analysis again, clear the check mark.

When you click **OK**, Capacity Planner closes the Data Point Exclusions dialog box and updates the chart to include the points removed from the list of exclusions.

Note: Data point exclusions are not saved with an analysis.

Showing or Hiding the Chart Legend

You can control whether or not the legend is displayed in an analysis chart by toggling the **Show/Hide Legend** button on the analysis window toolbar. When the legend is not shown, you can identify a line by clicking it, which displays the description of the line in the analysis window status bar.

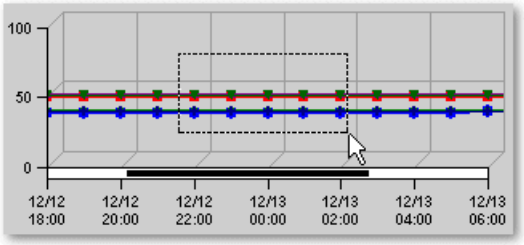
Zooming and Scrolling to View More or Less Chart Data

Depending on the date range and the number of data items and data sources you are analyzing, you may want to zoom in, zoom out, or scroll the contents of the chart window.

To view more detail or less detail, click the **Zoom In** and **Zoom Out** buttons on the Analysis Chart window toolbar.

You can also zoom in on a very specific part of the chart by dragging a rectangle around the area you want to magnify (Figure 4–8). Release the mouse button to zoom in on the selected area.

Figure 4–8 Zooming in on a Selected Area of the Chart

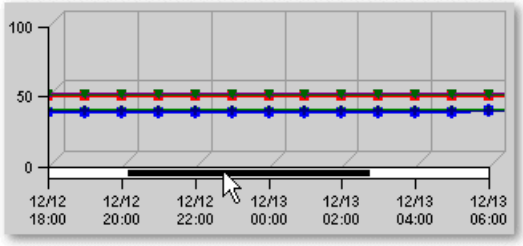


Often, the chart data will extend beyond the width of the Analysis Chart window. You can view the data that extends beyond the edges of the window by resizing the Analysis Chart window or by scrolling to the right or left.

To scroll the Analysis chart window:

1. Press and drag the pointer on the black scroll bar below the chart (Figure 4–9).
2. When the portion of the chart you want to view is displayed in the Analysis Chart window, release the mouse button.

Figure 4–9 Scroll Bar in the Analysis Chart Window



Auto-Scaling Data

If an analysis contains data from different classes where the scale of the data is significantly different, you may want to apply scaling to the data. Scaling normalizes the data so that all lines have a maximum value of no greater than 100. Very large data items are scaled down, and very small data items are scaled up.

To apply scaling, click **Toggle Auto Scaling** on the analysis window toolbar. When scaling is applied to a chart, data lines with very different scales may be visually correlated. When scaling is enabled, Capacity Planner displays the label **Scaled Data** along the Y-axis of the chart. If you select a line from the chart by clicking it, the scaling factor for that line is included in the status bar.

What Is a Trend Analysis?

A trend analysis infers or estimates future values by projecting and extending known values. Oracle Capacity Planner can perform the following types of trend analyses:

- Trend analysis to a point in time

  In a trend analysis to a point in time, you provide a goal time for a data item. Oracle Capacity Planner uses the specified values in the historical database to project the value of the data item at the goal time.

  A trend analysis to a point in time is used to answer questions such as, "How full will this disk be at the end of the year?"

- Trend analysis to a value

  In a trend analysis to a value, you provide a goal value for a data item. Oracle Capacity Planner uses the specified values in the historical database to project the time at which the data item will reach the goal value.

A trend analysis to a value is used to answer questions such as, "When will our CPU utilization reach 95 percent?"

Specifying and Performing a Trend Analysis

To specify and perform a trend analysis, follow these steps:

1. In the Analysis window, either:
   - Select a line by clicking a data point in the line, then click **Trend Analysis** on the toolbar.
   - Click the right mouse button on a data point in a line, then select **Trend Analysis** from the context menu.

   Both of these methods display the Trend Analysis dialog box.
2. In the **Date Range for Trend Analysis** section of the dialog box, select a date range to use as a basis for the trend analysis:
   - Click **Selected** to use the date range currently used for all the data in the chart (even data that extends beyond the edge of the chart window).
   - Click **Select Date Range** to select a different date range using the Select Date Range dialog box.

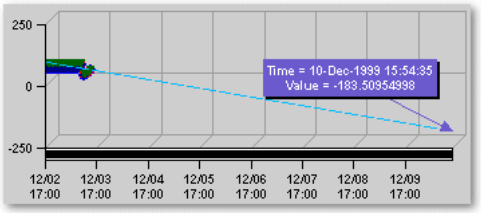
     For information about the fields on the Select Date Range dialog box, click **Help**.
   - Click **Visible** to use the range of dates currently visible in the analysis chart window (not including any data that extends beyond the edge of the chart window).

     This option is enabled only if you zoomed in to see more detail in the chart. For more information, see "Zooming and Scrolling to View More or Less Chart Data" on page 4-37.
3. In the Trend Analysis section of the dialog box, specify a date and time or a value, as follows:
   - To perform a trend analysis to a future date and time, click **Estimate the value at some date & time** and enter the time and date for which you wish to determine a value.
   - To perform a trend analysis to a value, click the **Estimate date for a specified value** and enter a numeric value for which you wish to determine a date and time.

4. Click **Annotate Trend** to display the Annotate Trend dialog box.

   Use the Annotate Trend dialog box to specify whether Capacity Planner will annotate the trend analysis (see Figure 4–10) and the appearance of the trend analysis line in the chart window. If you need information about the fields in the Annotate Trend dialog box, click **Help**.

Figure 4–10 Annotation for a Trend Analysis



5. Click **OK** to close the Annotate Trend dialog box and return to the Trend Analysis dialog box.
6. Click **Estimate** to calculate the result of the extrapolation analysis so the estimated value or date appears in the **Estimate** field.
7. Click either **OK** or **Apply** to update your analysis chart with the trend analysis results.

   If you click **Apply**, the chart is updated, but the Trend Analysis dialog box is not closed. The **Apply** button gives you the opportunity to view the results of the update and to fine-tune them if necessary.

Note: To move the annotation box in an analysis chart, click the box, hold the mouse button down, then move the mouse to a different position on the chart. Release the mouse button when you have reached the position in the chart where you want the annotation box displayed.

Saving Changes to a User-Defined or Predefined Analysis

You can save the changes you make to an analysis. Oracle Capacity Planner can determine if you are saving a user-defined analysis or a predefined analysis. The steps for saving your changes differ, depending on the type of analysis.

Saving the Results of a User-Defined Analysis

To save changes to a user-defined analysis:

1. While viewing the analysis, click **Save Chart** on the analysis window toolbar.
2. Click **Yes** in the message box.

   Your changes are saved to in the Analysis folder for the selected service using the current name of the analysis or data class. To save the analysis under a different name, choose **Save As** from the Analysis window **File** menu.

Saving the Results of a Predefined Analysis

If you modify a predefined analysis, you must provide a new name for the analysis when you save the results of the analysis. When you do this, you create a new user-defined analysis. Saving the analysis with a new name ensures that the original predefined analysis is not overwritten.

To save the results of a predefined analysis:

1. While viewing the analysis, click **Save Chart** on the analysis window toolbar.
2. Click **Yes** in the dialog box that is displayed.
3. In the Save Analysis As dialog box, type a unique name for the user-defined analysis that will be created.
4. Click **OK** in the Save Analysis As dialog box.

   Capacity Planner places the new user-defined chart in the Analyses folder of the navigator panel.

Printing an Analysis Chart

To print an analysis chart, click the **Print Chart** button in the Analysis Chart toolbar while the chart is displayed.

The size of the chart on the screen determines the size of the chart in the chart printout. For example, when you print a chart that is 5 inches high and 7 inches wide on your screen, the chart will be 5 inches high and 7 inches wide in the printout.

Generating a Report for an Analysis Chart

To generate an HTML report for an analysis chart:

1. Click the **Report Chart** button while the chart is displayed.

Capacity Planner displays a dialog box that tells you the name of the HTML file used to display the report and where the file will be saved.

2. Make a note of the file name and location.
3. Click **Yes** to preview the chart in your Web browser, or click **No** if you do not want to preview the report.

You can view the report later using a Web browser, or—if you have experience with Web publishing tools and techniques—you can share the report with your co-workers by publishing the HTML file and its associated graphic files on a Web server.

Understanding the Oracle Capacity Planner Historical Database

The logical structure of the Oracle Capacity Planner historical database schema was designed to facilitate querying of capacity planning data by tools other than Oracle Capacity Planner, such as Microsoft Excel. This section contains a brief description of the design of the Oracle Capacity Planner historical database schema. It is not intended to be a comprehensive description of all tables and columns in the schema.

Caution: Performing any modification to the Oracle Capacity Planner historical database schema (tables, columns, or indexes) or to the data itself is not recommended and is not supported. Such modifications may render your Oracle Capacity Planner historical database schema unusable as a target to load or retrieve capacity planning data.

Understanding the Service Reference Table (vp_service_ref)

The service reference table contains a list of all services for which data has been stored in the Oracle Capacity Planner historical database. The HOST column contains the name of the host where the Agent data gathering service that loaded data for the specified target (the NAME column) is located. The SVC_ID column is a column used for lookup into the class reference table, described in the following paragraph.

Understanding the Class Reference Table (vp_class_ref)

The class reference table contains a list of all collection classes for which data has been stored in the Oracle Capacity Planner historical database, for all services. To see the list of classes for a particular service, a query should look for the SVC_ID column equal to a SVC_ID value from the service reference table, described in the previous paragraph. Such a query would produce a list of all classes for which data exists in the database for a particular service.

Understanding the Data Tables

Each class of data in the class reference table has six data tables that contain the data for that collection class. Each table contains data for a particular time aggregate, for example, hours, days, weeks, months, and years. The base table contains the raw data as it is collected at the collection frequency. Therefore, the base table does not contain any statistical aggregations. The aggregate tables contain aggregate statistics for each data item collected for that class.

Data Table Names

Data tables are named according to the first four columns of the class reference table (in reverse order) plus the aggregate. Example 4–1 shows an entry for a collection class in the class reference table.

Example 4–1 Collection Class Entry in the Class Reference Table

```
CLS_ID      MOD_ID     SVC_ID     IS_UDS      CONTAINER_CLASS_LABEL
---------- ---------- ---------- ---------- ------------------------------
         8          2          2          0 I/O
```

The collection class in Example 4–1 would have the data tables shown in Example 4–2.

Example 4–2 Data Tables for a Collection Class

```
VP_0_2_2_8_BASE
VP_0_2_2_8_HOUR
VP_0_2_2_8_DAY
VP_0_2_2_8_WEEK
VP_0_2_2_8_MONTH
VP_0_2_2_8_YEAR
```

Data Table Contents

The first two columns of each data table are always the Timestamp column and the Data Source Name column. The Data Source Name column is essentially the class name, for example, Logical Disk. In aggregate tables (tables other than BASE), the third column is an accuracy calculation, which is not supported in this version of Oracle Capacity Planner.

All subsequent columns contain the values for the data items collected for that class, as they are actually named. In aggregate tables, these columns are prefixed with a two-character abbreviation for the aggregation statistic that the column contains. Table 4–4 shows these abbreviations and their meanings.

Table 4–4 Prefixes Used for Column Names in Aggregate Tables

Prefix	Meaning
AV	Average
SM	Sum
MN	Minimum
MX	Maximum
SD	Standard deviation
VR	Variance
CT	Count

These statistics are calculated for the set of data in that aggregate from the next lowest level. For example, a row in the DAY aggregate table contains statistics calculated for all entries in the HOUR table for that day.

5

Using Oracle TopSessions

This chapter covers how you use Oracle TopSessions, including the following topics:

Before You Start Oracle TopSessions

Before using Oracle TopSessions, it is best to ensure that the following views exist on the database instance:

- V$SESSION
- V$STATNAME
- V$SESSTAT
- ALL_TAB_COLUMNS
- V$OPEN_CURSOR
- V$SQLTEXT
- V$LOCK

- SYS.DBA_OBJECTS
- V$ROLLNAME
- PLAN_TABLE
- ALL_CATALOG
- AUDIT_ACTIONS

The first four tables of the list are essential; Oracle TopSessions cannot run without them. The rest of the tables are required to use certain features of Oracle TopSessions; however, you can still run Oracle TopSessions without them.

The smptsi*xx*.sql scripts have been provided by Oracle Enterprise Manager to create the required tables and views on each database you want to connect to from Oracle TopSessions. The *xx* in the file name identifies the version of the database against which the script should be run. For example, if you have an Oracle 7.3 database, you would run the smptsi73.sql script against it, and if you have an Oracle 8.0 database, you would run the smptsi80.sql script against it. The script for each database version is located in the $ORACLE_HOME\SYSMAN\ADMIN directory. Run these scripts from the SYS account. For more information about running these scripts, see "Setting Up TopSessions" on page 2-5.

If the ALL_CATALOG view or the AUDIT_ACTIONS view does not exist on a database, run catalog.sql against the database from the SYS account. This script is located in the $ORACLE_HOME\RDBMS*xx*\ADMIN directory.

Starting Oracle TopSessions

You start Oracle TopSessions as you would any typical Oracle Enterprise Manager application.

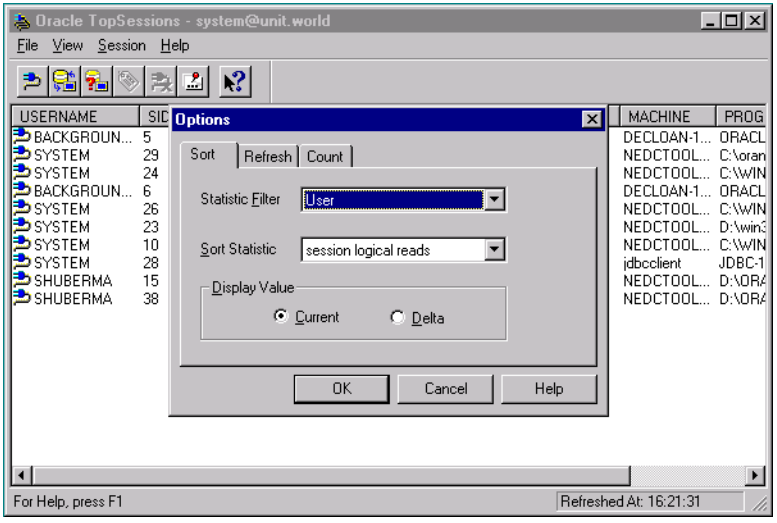
Attention: See "Before You Start Oracle TopSessions" on page 5-1 if an error message appears when you attempt to start Oracle TopSessions.

By default, the top *n* sessions in the main display are sorted by the session logical reads statistic. If you want to change the sort statistic and / or the statistic filter, choose Options on the Session menu. This displays the Options property sheet, on which you can customize the session information displayed in the main display. See "Customizing the Session Information You Display" on page 5-17 for more information about using the Options property sheet.

Note: The options that were selected when you last exited Oracle TopSessions are automatically loaded the next time you start Oracle TopSessions. These options are saved in ORACLE_HOME\SYSMAN\TEMP\TOPSESS.OPT.

Figure 5–1 shows the Oracle TopSessions main window and the Options property sheet.

Figure 5–1 Oracle TopSessions Main Window



If you are monitoring multiple databases, it is possible to start multiple Oracle TopSessions connections to multiple database instances. The system resources of the management console are the only constraining factors.

Oracle TopSessions Main Window

As Figure 5–1 shows, the Oracle TopSessions main window includes the following components:

- Title bar
- Menu bar

- Toolbar
- Main display
- Status bar

The following sections describe each of these components.

Title Bar

The title bar of the Oracle TopSessions main window displays the name of the application and the database instance to which an Oracle TopSessions connection has been established. If no connection to a database instance currently exists, the message "No database connection" is displayed in the title bar.

Toolbar

The toolbar of the Oracle TopSessions main window contains icons representing certain Oracle TopSessions menu items, including the following: Change Database Connection, Refresh, Toggle Manual/Auto Refresh, Session Details, Kill Session, Options, and Help. If no Oracle TopSessions connection to a database instance currently exists, all tools except the Change Database Connection and Help tools are disabled. If a database connection does exist, the Change Database Connection, Refresh, Toggle Manual/Auto Refresh, Options, and Help tools are enabled.

The Refresh Mode tool performs the same function as the Manual and Automatic buttons on the Refresh page of the Options property sheet. When the Refresh Mode tool is not pressed (default), manual refresh mode is enabled. When the Refresh Mode tool is pressed, automatic refresh mode is enabled.

Enabling automatic refresh mode with the Refresh Mode tool enables whatever refresh interval is set on the Refresh page of the Options property sheet. To adjust

this interval, go to the Refresh page itself. See "Customizing the Session Information You Display" on page 5-17 for more information.

> **Attention:** The length of time a data refresh requires varies, depending on the number of sessions connected to the database instance. A database instance with 700 to 800 sessions logged on would require about 10 seconds to refresh. Thus, an instance with perhaps 1,000 sessions logged on, when automatically refreshed at the default 10 second interval, would be in perpetual refresh mode. Thus, if you plan to use the automatic refresh mode, it is very likely that you will want to adjust the refresh interval to reflect your particular environment.

Status Bar

The status bar of the Oracle TopSessions main window can include the following information:

Left section

When a menu item has been selected, an explanation of the menu item appears in this section of the status bar.

Right section

Displays the time of the last data refresh of the Oracle TopSessions main display.

Oracle TopSessions Menu Bar

The menu bar of the Oracle TopSessions main window includes the following pull-down menus:

- File
- View
- Sessions
- Help

File Menu

The File menu items allow you to create new database connections and exit Oracle TopSessions.

The File menu includes the following commands:

Change Database Connection

Displays the Login Information dialog box, enabling you to connect to a database instance of your choice.

Exit

Exits Oracle TopSessions. See "Exiting Oracle TopSessions" on page 5-20 for more information.

View Menu

The View menu items allow you to refresh the main window display and to show / hide the toolbar or statusbar.

The View menu includes the following commands:

Refresh

If manual refresh mode (default) is enabled, you can choose this command to refresh (immediately) the session information displayed in the Oracle TopSessions main window.

Toolbar

Allows you to show / hide the toolbar.

Status Bar

Allows you to show / hide the statusbar.

Sessions Menu

The Sessions menu items allow you to specify what session statistics will be displayed in the Oracle TopSessions main display, and how that information will be refreshed. See "Customizing the Session Information You Display" on page 5-17 for more information about these menu items.

The Sessions menu includes the following commands:

Details

Displays the Session Details window for the session selected from the Oracle TopSessions main display. This menu item is disabled when no session in the Oracle TopSessions main display has been selected.

Kill

Kills the user session selected on the Oracle TopSessions main display.

When a session has been killed, a red "X" marks the username of that session. This symbol is like the red "X" on the Kill tool icon.

Options

Displays the Options property sheet, from which you can specify the sorting criterion, the refresh mechanism, and the number of entries for statistics displayed in the Oracle TopSessions main window.

The length of time the data refresh requires varies, depending on the number of sessions connected to the database instance. An instance with 700 to 800 sessions logged on would require about 10 seconds to refresh.

If no Oracle TopSessions connection to a database instance currently exists, all of these Session menu items are disabled.

Help Menu

The Help menu items allow you to obtain help on Oracle TopSessions menu items or property sheets.

Contents

Displays an overview of the Oracle TopSessions help system.

Search for Help On

Displays an alphabetical list of Help topics.

Using Help

Displays information about using the Help system.

About Oracle TopSessions

Displays the version number of this application.

Obtaining an Overview of Session Activity

Once populated with data, the Oracle TopSessions main display includes a multi-column list of summary information for each session connected to the database instance, or for each of the top *n* sessions, as measured by the selected sort statistic. This information is obtained from the V$SESSION view, as follows:

SID

Oracle session ID for this session.

USERNAME

Oracle user name using this session.

OSUSER
Operating system user name.

Sort Statistic
This fourth column varies, depending on the sort statistic you have selected.

COMMAND
Last Oracle command executed by this session.

STATUS
Status of this session: IDLE, ACTIVE, KILLED, or BLOCKED.

MACHINE
Identifier of the machine using this session.

PROGRAM
Name of the client program executing this session.

See "Customizing the Session Information You Display" on page 5-17 for information about defining the top *n* sessions.

For information on the V$ views from which these statistics are derived, see the *Oracle Server Reference*. For information on how to make use of these statistics, see *Oracle Server Tuning* and *Oracle Server Administrator's Guide*.

Using the Right Mouse Button on the Main Display

In the Oracle TopSessions main display, you can click on a username with the right mouse button, and then select one of the following menu items:

Details
Displays the Session Details window for that session. See "Viewing Details About a Given Session" on page 5-9 for more information.

Kill
Kills the selected session. This menu item works like the Kill menu item of the Session menu.

Sorting the Order of Entries in the Main Display

By default, the value of the sort statistic determines the order of session entries in the Oracle TopSessions main display. (The sort statistic is the fourth column of the

multi-column list.) However, you can use any field in the list to sort the order of displayed session entries, by clicking on any column header in the list.

Viewing Details About a Given Session

To obtain more information about a given session, you can drill down by double clicking on an SID in the Oracle TopSessions main display. A Session Details window for that session appears. This section covers the following topics:

- Session Details pages
- Session Details data refresh time
- Viewing session details for multiple sessions simultaneously

Session Details Pages

The Session Details window includes the following pages:

- General
- Statistics
- Cursors
- Locks

To select one of these pages, click on the appropriate tab title near the top of the Session Details window.

Session Details General Page

The General page of the Session Details window provides detailed information about a session. This information is obtained from the V$SESSION view.

For information on the V$ views from which these statistics are derived, see the *Oracle Server Reference*. For information on how to make use of these statistics, see *Oracle Server Tuning* and *Oracle Server Administrator's Guide*.

While the contents of the General page can vary, depending on the Oracle7 or Oracle8 server release, for an Oracle 7.3 server, the General page contains the following:

SADDR
Session address of this session.

SID
Oracle session ID of this session.

SERIAL#
Serial number of this session. Together with the SID, this number provides a unique key for this session.

AUDSID
Auditing session ID.

PADDR
Address of the Oracle process using this session.

USER#
ID of the user using this session.

USERNAME
User name of the user using this session.

COMMAND
Last Oracle command executed.

TADDR
Address of the transaction state object.

LOCKWAIT
Address of lock waiting for; null if none.

STATUS
Status of this session: ACTIVE, IDLE, KILLED, or BLOCKED.

SERVER
Server type: DEDICATED, SHARED, PSEUDO, or NONE.

SCHEMA#
Schema user identifier.

SCHEMANAME
Schema user name.

OSUSER
Operating system user name.

PROCESS
Process ID.

MACHINE
Identifier of the machine using this session.

TERMINAL
Operating system terminal name.

PROGRAM
Name of the client program using this session.

TYPE
Session type.

SQL_ADDRESS
Together with the SQL hash value, identifies the SQL statement that is currently being executed.

SQL HASH VALUE
Together with the SQL address, identifies the SQL statement that is currently being executed.

PREV_SQL_ADDR
Together with the previous SQL hash value, identifies the SQL statement that executed previous to the SQL statement currently executing.

PREV_HASH_VALUE
Together with the previous SQL address, identifies the SQL statement that executed previous to the SQL statement currently executing.

MODULE
Name of the currently executing module as set by calling the procedure named DBMS_APPLICATION_INFO.SET_MODULE.

MODULE_HASH
The hash value of the above module name.

ACTION
Name of the currently executing action as set by calling the procedure named DBMS_APPLICATION_INFO.SET_ACTION.

ACTION_HASH
The hash value of the above action name.

CLIENT_INFO
Information set by the DBMS_APPLICATION_INFO.SET_CLIENT_INFO procedure.

FIXED_TABLE_SEQUENCE
Number that increases every time the session completes a call to the database and there has been an intervening select from a dynamic performance table.

ROW_WAIT_OBJ#
Object ID for the table containing the row ID specified in ROW_WAIT_ROW#.

ROW_WAIT_FILE#
Identifier for the datafile containing the row ID specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is non-zero.

ROW_WAIT_BLOCK#
Identifier for the block containing the row ID specified in ROW_WAIT_ROW#. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is non-zero.

ROW_WAIT_ROW#
Current row ID being locked. This column is valid only if the session is currently waiting for another transaction to commit and the value of ROW_WAIT_OBJ# is non-zero.

LOGON_TIME
Logon timestamp for the session.

LAST_CALL_ET
Last call made during the session.

Session Details Statistics Page

The Statistics page of the Session Details window provides a large number of performance statistics for the selected session. The particular statistics this page displays can vary, depending on the Oracle server instance. These statistics are obtained from the V$SESSTAT view.

The selected category in the Category drop-down menu determines the statistics that will be displayed for the current session. To define a custom set of statistics to display, click the Custom... button on the Statistics page. Then use the Choose Statistics for Display dialog box to select the set of statistics. After you have chosen a set of statistics, they are saved as the Custom category of statistics and the Custom category is added to the Category drop-down menu. Then, to select the Custom category of statistics for display, select Custom from the Category drop-down menu. See "Choose Statistics for Display Dialog Box" on page 5-13 for more information on choosing a custom set of statistics.

For information about the statistics that can be displayed for a session and the V$ views from which they are derived, see the *Oracle Server Reference*. For information on how to make use of these statistics, see *Oracle Server Tuning*.

Choose Statistics for Display Dialog Box

The Choose Statistics for Display dialog box allows you to select a custom set of statistics to display for a session.

The left tree view contains the list of statistics that are currently selected for display in the Custom category on the Statistics page of the Session Details window. The right tree view contains the available statistics that can be selected for display in the Custom category. In the tree views, a + or - sign icon is used for the names of the categories of statistics. Click the + or - sign for a category of statistics to view or hide the individual statistics in the category.

To move a category of statistics or an individual statistic from one tree view to the other, double-click on the category or statistic. Click OK when the left tree view contains the set of statistics you want to display in the Custom category. After you have chosen a set of statistics, they are saved as the Custom category of statistics and the Custom category is added to the Category drop-down menu on the Statistics page of the Session Details window. Then, to select the Custom category of statistics for display, select Custom from the Category drop-down menu on the Statistics page of the Session Details window.

You can modify the set of statistics in the Custom category by clicking the Custom... button on the Statistics page of the Session Details window, then selecting a different set of statistics using the Choose Statistics for Display dialog box.

The set of statistics in the Custom category are saved in $ORACLE_HOME\SYSMAN\TEMP\TOPSESSC.INI.

Session Details Cursors Page

The Cursors page of the Session Details window provides information about each shared cursor in the shared SQL area for the selected session.

> **Attention:** In order to view explain plans for SQL statements accessing V$ views, you must log in as SYS. In addition, you must have privileges to access the objects in the SQL statement for which you want to view an explain plan.

The Cursors page contains the following:

Show Cursors

Click on an item in this list box to display SQL statement(s) for the selected user session. Options include:

Current Cursor: Displays the SQL statement currently executing. This is the default.

All Cursors: Displays all SQL statements that have executed or will execute on the database server for the selected user session.

Show Explain Plan

Click on this button to display the explain plan for the SQL statement in the Cursors page display box which you have selected. See "Displaying an Explain Plan for a User Session" on page 5-15 for more information.

Display box

Displays one or more SQL statements for the selected user session. If All Cursors has been selected, the first SQL statement in the list is the statement currently executing on the server for the session.

> **Attention:** If no SQL statement appears in this display box, but you expect one to exist for the session, a problem may exist with an important database table. See "Before You Start Oracle TopSessions" on page 5-1 for more information.

> **Attention:** Only SQL statements containing the commands SELECT, INSERT, UPDATE and DELETE will produce explain plans for you to view in this display box; any other SQL statement will not produce an explain plan for you to view.

Displaying an Explain Plan for a User Session

To display an explain plan for a user session, from the Cursors page of the Session Details window, select a SQL statement and click on the Show Explain Plan button. The Explain Plan For Session *xx:y* window appears.

The title bar of this window includes the SID *(xx)* of the session and the number of windows opened on that session *(y)*. This window includes the following elements.

SQL Statement
Displays the SQL statement for which you want to view the explain plan.

Explain Plan
The folders allow you to collapse or expand the level of detail of the explain plan. This feature is particularly useful when analyzing long and / or complicated explain plans.

> **Note:** The Expected Rows column only appears when an Oracle 7.3 database is being monitored.

Session Details Locks Page

The Locks page of the Session Details display provides information about locks held or requested by a given session. This information is obtained from the V$LOCK and V$ROLLNAME views.

For information about the V$ views from which these statistics are derived, see the *Oracle Server Reference*. For information on how best to interpret and use these statistics, see *Oracle Server Tuning*.

The Locks page includes the following list box:

Lock Types
Clicking on an item in this list box allows you to select the type of lock by which the sessions will be displayed. Options include All Locks or Blocking / Waiting Locks.

The Locks page contains the following fields:

User Name
Oracle user name using the session.

Session ID
Oracle session ID for the session.

Lock Type
Type of lock, as follows: MR (Media Recovery); RT (Redo Thread); UN (User Name); TX (Transaction); TM (DML); UL (PL/SQL User Lock); DX (Distributed Xaction); CF (Control File); IS (Instance State); FS (File Set); IR (Instance Recovery); ST (Disk Space Transaction); TS (Temp Segment); IV (Library Cache Invalidation); LS (Log Start or Switch); RW (Row Wait); SQ (Sequence Number); TE (Extend Table); and TT (Temp Table).

Mode Held
Mode in which the lock is currently held by the session, as follows: None; Null; Row-S (SS); Row-X (SX); Share; S/Row-X (SSX); and Exclusive.

Mode Requested
Mode in which the lock is being requested by the process, as follows: None; Null; Row-S (SS); Row-X (SX); Share; S/Row-X (SSX); and Exclusive.

Object Name
Name of the object, rollback segment, table or view, being locked. If the lock type is TM, the object is a table or view. If the lock type is TX, the object is a rollback segment.

Object Owner
Owner of the object that has been locked by the session. The session user, listed in the Username field of the Session Details window General page, may be different than the owner of the object being locked.

Object Type
Object type.

Object ID
Unique identifier of the object.

Resource ID 1
For certain types of locks, this value is the object ID or rollback segment number.

Resource ID 2
Undocumented.

Attention: If no information is displayed on the Locks page, it is likely that no locks currently exist for the session. However, if you suspect that locks exist, but no information is displayed on the Locks page, a problem may exist with an important database table. See "Before You Start Oracle TopSessions" on page 5-1 for more information.

Session Details Display Refresh Time

Like the status bar of the Oracle TopSessions main display, the status bar of the Session Details display also shows the refresh time for the information it displays. This time reflects the refresh time of the Oracle TopSessions main display at the time the Session Details display is created. Even when the Oracle TopSessions main display is subsequently refreshed, the information in a Session Details window that remains open during the main display refresh will not itself be refreshed.

Likewise, even though the refresh time in the status bar of the main window is updated to reflect the most recent refresh, the refresh time in the status bar of an open Session Details window is not updated simultaneously. Thus, the refresh time in the Session Details window status bar continues to reflect the time the static information in the Session Details window was captured.

Should you want to refresh the information for a given session in the Session Details window, close the window and then double-click on the SID for that session in the Oracle TopSessions main window. The resulting Session Details window will reflect the latest refresh time of the Oracle TopSessions main display.

Viewing Session Details for Multiple Sessions

You can display, minimize (iconify), and maximize Session Details displays for multiple sessions as you track down problems and work to resolve them.

Customizing the Session Information You Display

To customize how session information is displayed in the Oracle TopSessions main window, choose Options from the Session menu. The Options property sheet that appears includes the following pages:

- Sort

- Refresh
- Count

To select one of these pages, click on the appropriate tab title near the top of the Options property sheet.

> **Note:** The options that were selected when you last exited Oracle TopSessions are automatically loaded the next time you start Oracle TopSessions. These options are saved in ORACLE_HOME\SYSMAN\TEMP\TOPSESS.OPT.

Options Sort Page

Use the Sort page of the Options property sheet to specify how session information displayed in the Oracle TopSessions main window will be sorted, and what time interval the data will reflect. The Sort page contains the following:

Statistic Filter

Click on an item in this list box to specify the group of statistics from which to then select a particular sort statistic in the Sort Statistic list box.

These groups of statistics are labeled as follows: Predefined, User (default), Redo, Enqueue, Cache, Operating System, Parallel Server, SQL, Debug, Other, and All.

Selecting one of these items makes available a different group of statistics in the Sort Statistics box. With the exception of the Predefined group, the mapping between a statistic and its statistic filter group is derived from the CLASS column of the dynamic performance table V$STATNAME.

The Predefined statistics are defined or calculated from statistics displayed on the Statistics page of the Session Details display, as follows: CPU Usage (CPU used by this session); File I/O (physical reads + physical writes); Memory (session uga memory); Open Cursors (opened cursors current); User Transactions (user commits + user rollbacks).

Sort Statistic

Click on this list box to select the statistic by which the sessions displayed in the Oracle TopSessions main window will be sorted. The contents of this list box vary, depending on the Statistic Filter that has been selected.

Display Value

Current: Click on this button to specify that statistics displayed in the Oracle TopSessions main window be based on the most recent data refresh.

Delta: Click on this button to specify that displayed statistics reflect the difference (delta value) between the most recent data refresh and the data refresh previous to that.

Options Refresh Page

Use the Refresh page of the Options property sheet to specify how you want session information displayed in the Oracle TopSessions main window to be refreshed. The length of time the data refresh requires will vary, depending on the number of sessions connected to the database instance. An instance with 700 to 800 sessions logged on would require about 25 seconds to refresh.

The Refresh page contains the following:

Manual

Click on this button (default) to enable manual refreshing of the session information displayed in the Oracle TopSessions main window. You can then use the Refresh command from the View menu or the Refresh tool to refresh this information when you so choose.

Automatic

Click on this button to enable automatic refresh of session information displayed in the Oracle TopSessions main window; the Refresh Interval group of list boxes is then enabled for use.

Refresh Interval

Click on the appropriate combination of the Seconds, Minutes and Hours list boxes to specify the refresh interval for session information displayed in the Oracle TopSessions main window. The default refresh interval is 10 seconds.

Attention: The refresh interval is the time interval measured from the beginning of a refresh to the beginning of the refresh following it. Thus, a database instance with perhaps 1,000 sessions logged on, and being automatically refreshed at the default 10 second interval, would be in perpetual refresh mode. Thus, if you plan to use the Automatic refresh mode, it is very likely that you will want to adjust the refresh interval to reflect your particular environment.

Options Count Page

Use the Count page of the Options property sheet to specify the number of sessions to be displayed in the Oracle TopSessions main window. The Count page contains the following:

Display All Sessions
Click this button to display all sessions connected to the database instance. As this option is very resource-intensive in almost all cases, Oracle Corporation recommends that you use the default option Display Top N Sessions.

Display Top N Sessions
Click on this button (default) to display the top *n* sessions that meet the sort statistic criterion specified in the Sort page of the Options property sheet. The default number of sessions is 10. Click on the scroll box to the right of this button to specify the value of *n*.

Exiting Oracle TopSessions

To exit Oracle TopSessions, choose Exit from the File menu. A dialog box is displayed, confirming that you want to close your Oracle TopSessions session.

6

Oracle Advanced Event Tests

The Event Management System (EMS) within Oracle Enterprise Manager assists the DBA with automatic problem detection and correction. Using the EMS, the DBA can establish boundary thresholds for warnings and alerts conditions within the network environment for problem monitoring.

The Oracle Enterprise Manager console provides event management functionality with several basis events. To use the Oracle Advanced Event Tests, you must have the licensed Oracle Diagnostics Pack.

The Oracle Advanced Event Tests for the database, listener, and node service types are grouped into the following categories:

- Database Fault Management Event Tests
- Database Space Management Event Tests
- Database Resource Management Event Tests
- Database Performance Management Event Tests
- Database Audit Management Event Test
- Node Fault Management Event Tests
- Node Space Management Event Tests
- Node Performance Management Event Tests

Note: For more information on the basic event tests or for information on using the Oracle Enterprise Manager Event Management System, see the *Oracle Enterprise Manager Administrator's Guide*.

Database Fault Management Event Tests

This category of event tests monitors for severe problems that require immediate action.

Database Alert

This event test signifies that the database being monitored has generated errors to the ALERT log file since the last sample time. The ALERT log file is a special trace file containing a chronological log of messages and errors. An alert event is triggered when Oracle Exception (ORA-006xx), deadlock detected (ORA-00060), or data block corrupted (ORA-01578) messages are written to the ALERT log file. A warning is displayed when other ORA messages are written to the ALERT log file.

Parameters
None

Output
Alert log error messages since last sample time.

Default Frequency
60 seconds

User Action
Examine ALERT log for additional information.

Archiver Hung

This event test signifies that the archiver of the database being monitored has been temporarily suspended since the last sample time.

If the database is running in ARCHIVELOG mode, an alert is displayed when archiving is hung (ORA-00257) messages are written to the ALERT file. The ALERT file is a special trace file containing a chronological log of messages and errors.

If the database is not running in ARCHIVELOG mode, this event will not register.

Parameters
None

Output
ALERT log error messages since last sample time.

Default Frequency
60 seconds

User Action
Examine ALERT log and archiver trace file for additional information; however, the most likely cause of this message is that the destination device is out of space to store the redo log file. Verify the device specified in the initialization parameter ARCHIVE_LOG_DEST is set up properly for archiving.

Broken Jobs

The Oracle server job queue is a database table that stores information about local jobs such as the PL/SQL call to execute for a job such as when to run a job. Database replication is also managed by using the Oracle job queue mechanism using jobs to push deferred transactions to remote master sites, to purge applied transactions from the deferred transaction queue or to refresh snapshot refresh groups.

A job can be broken in two ways:

1. Oracle has failed to successfully execute the job after sixteen attempts
2. The job has been explicitly marked as broken by using the procedure DBMS_JOB.BROKEN

This event test checks for broken DBMS jobs. An alert is generated if the number of broken jobs exceeds the value specified by the threshold argument.

Parameters
Alert threshold: Threshold for alert (number of jobs). Default is 0 jobs.

Output
Job identifiers of broken DBMS jobs.

Default Frequency
60 seconds

User Action
Check the ALERT log and trace files for error information. Correct the problem that is preventing the job from running. Force immediate re-execution of the job by calling DBMS_JOB.RUN, if desired.

Data Block Corruption

This event test signifies that the database being monitored has generated a corrupted block error to the ALERT file since the last sample time. The ALERT file is a special trace file containing a chronological log of messages and errors. An alert event is triggered when data block corrupted (ORA-01578) messages are written to the ALERT file.

Parameters
None

Output
Segment name containing the data block specified by the file ID and block number in the ORA-01578 message.

Default Frequency
60 seconds

User Action
Examine the ALERT log for additional information.

Database UpDown

This event test checks whether the database being monitored is running. If this test is triggered, other database events are ignored.

Parameters
None

User Action
The Startup Database job task can be set up as a fixit job for automatically correcting the problem.

Note: If the listener serving a database is down, this event may be triggered because the Intelligent Agent uses the listener to communicate with the database. This note applies to Intelligent Agent's released before 8.0.5.

Deferred Transactions

Oracle uses deferred transactions to propagate data-level changes asynchronously among master sites in an advanced replication system as well as from an updatable snapshot to its master table.

This event test checks for the number of deferred transactions. An alert is generated if the number of deferred transactions exceeds the value specified by the threshold argument.

Parameters

Threshold for alert (number of transactions). Default is 100 transactions.

Output

Number of deferred transactions.

Default Frequency

60 seconds

User Action

When the advanced replication facility pushes a deferred transaction to a remote site, it uses a distributed transaction to ensure that the transaction has been properly committed at the remote site before the transaction is removed for the queue at the local site. If transactions are not being pushed to a given remote site, verify that the destination for the transaction was correctly specified. If you specify a destination database when calling DBMS_DEFER_SYS.SCHEDULE_EXECUTION using the DBLINK parameter or DBMS_DEFER_SYS.EXECUTE using the DESTINATION parameter, make sure the full database link is provided.

Wrong view destinations can lead to erroneous deferred transaction behavior. Verify the DEFCALLEST and DEFTRANDEST views are the definitions from the CATREPC.SQL and not from the CATDEFER.SQL.

Error Transactions

Oracle uses deferred transactions to propagate data-level changes asynchronously among master sites in an advanced replication system as well as from an updatable snapshot to its master table. If a transaction is not successfully propagated to the remote site, Oracle rolls back the transaction, logs the transaction in the SYS.DEFERROR view in the remote destination database.

This event test checks for the number of transactions in SYS.DEFERROR view and raises an alert if it exceeds the value specified by the threshold argument.

Parameters
Threshold for alert (number of error transactions). Default is 0 transactions.

Output
Number of transactions that could not be applied

Default Frequency
60 seconds

User Action
An error in applying a deferred transaction may be the result of a database problem, such as a lack of available space in the table that is to be updated or may be the result of an unresolved insert, update, or delete conflict. The SYS.DEFERROR view provides the ID of the transaction that could not be applied. Use this ID to locate the queued calls associated with the transaction. These calls are stored in the SYS.DEFCALL view. You can use the procedures in the DBMS_DEFER_QUERY package to determine the arguments to the procedures listed in the SYS.DEFCALL view.

Failed Jobs

The Oracle server job queue is a database table that stores information about local jobs such as the PL/SQL call to execute for a job such as when to run a job. Database replication is also managed by using the Oracle job queue mechanism using jobs to push deferred transactions to remote master sites, to purge applied transactions from the deferred transaction queue or to refresh snapshot refresh groups.

If a job returns an error while Oracle is attempting to execute it, the job fails. Oracle repeatedly tries to execute the job doubling the interval of each attempt. If the job fails sixteen times, Oracle automatically marks the job as broken and no longer tries to execute it.

This event test checks for failed DBMS jobs. An alert is generated if the number of failed jobs exceeds the value specified by the threshold argument.

Parameters
Alert threshold: Threshold for alert (number of jobs). Default is 0 jobs.

Output
Job identifiers of failed DBMS jobs.

Number of failures since last successful execution.

Default Frequency
60 seconds

User Action
Check the ALERT log and trace files for error information. Correct the problem that is preventing the job from running.

Probe

This event test checks whether a new connection can be established to a database. If the maximum number of sessions is exceeded or the listener is down, this test is triggered.

> **Note:** If the preferred user has the RESTRICTED SESSION privilege, the user will be able to connect to a database even if the LICENSE_MAX_SESSIONS limit is reached.

Parameters
None

Output
None

Default Frequency
60 seconds. However, a frequency of 10 minutes is recommended.

User Action
Check the status of the listener to make sure it is running on the node where the event was triggered. If the listener is running, check to see if the number of sessions is at the session limit.

Session Terminated

This event test signifies that a session terminated unexpectedly since the last sample time. The ALERT file is a special trace file containing a chronological log of messages and errors. An alert is displayed when session unexpectedly terminated (ORA-00603) messages are written to the ALERT file.

Parameters
None

Output
ALERT log error messages since the last sample time.

Default Frequency
60 seconds

User Action
Examine the ALERT log and the session trace file for additional information.

Unscheduled Jobs

The Oracle server job queue is a database table that stores information about local jobs.

This event test checks for unscheduled DBMS jobs. An alert is generated when the number of jobs, whose execution time has exceeded the value specified by the Job Completion Time argument, exceeds the value specified in the Alert Threshold. A job's completion date/time is calculated using the NEXT_DATE value in the SYS.DBA_JOBS view plus the approximate time it takes to complete a job as specified by the Job completion time argument.

Parameters
1. Alert threshold: Threshold for alert (number of jobs). Default is 0.
2. Job completion time: Approximate time for job completion in minutes. Default is 10 minutes.

Output
Job identifiers of jobs that are not rescheduled for execution.

Default Frequency
60 seconds

User Action
Check the FAILURES and BROKEN values in the SYS.DBA_JOBS view.

If the job failed to execute, check the ALERT log and trace files for error information and fix the error.

If the job was never executed, there may be a problem with the availability of SNP background processes. Check the initialization parameter JOB_QUEUE_PROCESSES to determine the maximum number of background processes available and JOB_QUEUE_INTERVAL to determine how frequently each background process wakes up.

User Blocks

This event test signifies that a database user is blocking at least one other user from performing an action, such as updating a table. An alert is generated if the number of consecutive blocking occurrences reaches the specified value.

> **Note:** The `catblock.sql` script needs to be run on the managed database prior to using the User Blocks test. This script creates some additional tables, views, and public synonyms that are required by the User Blocks test.

Parameters
Number of occurrences: Number of consecutive occurrences a user can be blocked before an alert is generated. Default is three.

Output
Session ID of the user who is blocking other users.

Default Frequency
60 seconds

User Action
Either have the user, who is blocking other users, rollback the transaction, or wait until the blocking transaction has been committed.

Database Space Management Event Tests

This category of event tests tracks possible space problems within the database.

ALERT File Large

The ALERT file is a special trace file containing chronological log of messages and errors. Oracle always appends to the file. To control the size of an ALERT file you must manually delete the file when you no longer need it.

This event test checks for file size of the ALERT file. If the file is greater than the values specified in the threshold arguments, then a warning or alert is generated.

Parameters

1. Alert threshold: Filesize in kilobytes for alert. Default 100,000 kilobytes.
2. Warning threshold: Filesize in kilobytes for warning. Default 50,000 kilobytes.

Output
Current size of ALERT file in kilobytes.

Default Frequency
60 seconds. However, a frequency of 10 minutes is recommended.

User Action
Delete the ALERT file to recover disk space.

> **Note:** The ALERT file can be safely deleted while the instance is running, although you might want to make an archived copy of it first.

Archive Full

When running a database in ARCHIVELOG mode, the archiving of the online redo log is enabled. Filled groups of the online redo log are archived, by default, to the destination specified by the LOG_ARCHIVE_DEST initialization parameter. If this destination device becomes full, the database operation is temporarily suspended until disk space is available.

If the database is running in ARCHIVELOG mode, this test checks for available redo log destination devices. If the space available is less than the threshold value given in the threshold arguments, then an alert or warning is generated.

If the database is not running in ARCHIVELOG mode, or all archive destinations are standby databases for Oracle 8i, this test fails to register.

Parameters
1. Alert threshold: Free space threshold in kilobytes for alert. Default 2000 kilobytes.
2. Warning threshold: Free space threshold in kilobytes for warning. Default 5000 kilobytes.

Output
Space available on destination drive in kilobytes.

> **Note:** If you have more than one number for the amount of free space available, this means you have more than one destination. Check the amount of free space for **all** destinations.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

Verify the device specified in the initialization parameter LOG_ARCHIVE_DEST is set up properly for archiving.

- For Oracle7, verify that the LOG_ARCHIVE_DEST initialization parameter is set up properly for archiving.
- For Oracle8, verify that the LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST initialization parameters are set up properly for archiving.
- For Oracle8i, there are two methods you can use to specify archive destinations. The first method is to use the LOG_ARCHIVE_DEST_*n* parameter (where *n* is an integer from 1 to 5) to specify from one to five different destinations for archival. Each numerically-suffixed parameter uniquely identifies an individual destination, for example, LOG_ARCHIVE_DEST_1, LOG_ARCHIVE_DEST_2, and so on. The second method, which allows you to specify a maximum of two locations, is to use the LOG_ARCHIVE_DEST parameter to specify a *primary* archive destination and the LOG_ARCHIVE_DUPLEX_DEST parameter to determine an optional *secondary* location.

Archive Full (%)

The Archive Full (%) event test monitors the same destination device as the Archive Full event test. The Archive Full (%) event test, however, returns the percentage of free space remaining on the log destination.

If the space available is less than the threshold value given in the threshold arguments, then an alert or warning is generated.

If the database is not running in ARCHIVELOG mode or all arachive destinations are standby databases for Oracle8i, this test fails to register.

Parameters

1. Alert threshold: Percentage of free space threshold for alert. Default is 30%.
2. Warning threshold: Percentage of free space threshold for warning. Default is 50%.

Output

Percentage of space available on destination drive.

Note: If you have more than one number for the amount of free space available, this means you have more than one destination. Check the amount of free space for **all** destinations.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

Verify the device specified in the initialization parameter LOG_ARCHIVE_DEST is set up properly for archiving.

- For Oracle7, verify that the LOG_ARCHIVE_DEST initialization parameter is set up properly for archiving.
- For Oracle8, verify that the LOG_ARCHIVE_DEST and LOG_ARCHIVE_DUPLEX_DEST initialization parameters are set up properly for archiving.
- For Oracle8i, there are two methods you can use to specify archive destinations. The first method is to use the LOG_ARCHIVE_DEST_*n* parameter (where *n* is an integer from 1 to 5) to specify from one to five different destinations for archival. Each numerically-suffixed parameter uniquely identifies an individual destination, for example, LOG_ARCHIVE_DEST_1, LOG_ARCHIVE_DEST_2, and so on. The second method, which allows you to specify a maximum of two locations, is to use the LOG_ARCHIVE_DEST parameter to specify a *primary* archive destination and the LOG_ARCHIVE_DUPLEX_DEST parameter to determine an optional *secondary* location.

Chunk Small

The Oracle Server allocates space for segments in units of one extent. When the existing extents of a segment are full, the Oracle Server allocates another extent for that segment. In order to do so, Oracle searches through the free space in the tablespace containing the segment for the first free, contiguous set of data blocks sufficient to meet the required extent's size. If sufficient space is not found, an error is returned by the Oracle Server.

This event test checks for the largest chunk of free space in the tablespace specified by the Tablespace name, Segment name, and Segment type parameters. If any table, index, cluster or rollback segments within the tablespace cannot allocate the additional number of extents specified in the thresholds, then a warning or alert is generated.

Example

If the largest chunk of free space in the specified tablespace can only contain 2 extents, then 2 is compared to the threshold values. If 3 is specified for an alert, the alert test is triggered because 3 extents cannot be allocated in the tablespace.

Parameters

1. Tablespace name filter: Filter of the tablespace names to be monitored, or * for all tablespaces. Tablespaces that are either temporary, read-only, or off-line are excluded. Default is *.
2. Segment name filter: Filter of the segments to be monitored, or * for all segments. Default is *.
3. Segment type filter: Filter of the segment types to be monitored, or * for all segment types. Segment types CACHE and DEFERRED ROLLBACK are excluded.
4. Alert threshold: Threshold for alert (number of extents). Default extent is 1.
5. Warning threshold: Threshold for warning (number of extents). Default is 2 extents.

Output

Segment name where the additional extents cannot be allocated.

Tablespace name containing the segment.

Maximum size of contiguous free space in kilobytes for the tablespace.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

Increase the size of the tablespace by enabling automatic extension for one of its existing datafiles, manually resizing one of its existing datafiles or adding a new datafile.

Or if the tablespace is suffering from tablespace free space fragmentation problems, consider reorganizing the entire tablespace by dropping and recreating all segments within that tablespace. When reorganizing a tablespace, consider making the extents to be sized as integral divisors of the usable size of the datafiles in which they reside. Try to limit the extent sizes used in the tablespace to be no more than 2 or 3 different extent sizes. Ensure extents within a segment are the same size or a multiple of each other by specifying STORAGE parameters where NEXT=INITIAL

and PCTINCREASE=0. For segments that are linearly scanned, choose an extent size that is a multiple of the number of blocks read during each multiblock read.

Dump Full

Each server and background process can write to an associated trace file in order to log messages and errors. Background processes and the ALERT file are written to the destination specified by BACKGROUND_DUMP_DEST.

Trace files for server processes are written to the destination specified by USER_ DUMP_DEST.

This event test checks for available free space on these dump destination devices. If the space available is less than the threshold value given in the threshold arguments, then an alert or warning is generated.

Parameters

1. Alert threshold: Free space threshold in kilobytes for alert. Default 2000 kilobytes.
2. Warning threshold: Free space threshold in kilobytes for warning. Default 5000 kilobytes.

Output

Dump destination device and space available in kilobytes.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

Verify the device specified in the initialization parameters BACKGROUND_ DUMP_DEST and USER_DUMP_DEST is set up properly for tracing.

Dump Full (%)

This event test monitors the same dump destinations as the Dump Full event test. The Dump Full (%) event test, however, returns the percentage of free space remaining on the dump destinations.

If the space available is less than the threshold value given in the threshold arguments, then an alert or warning is generated.

Parameters

1. Alert threshold: Percentage of free space for alert. Default 30%.

2. Warning threshold: Percentage of free space for warning. Default 50%.

Output
Dump destination device and percentage of space available.

Default Frequency
60 seconds. However, a frequency of 10 minutes is recommended.

User Action
Verify the device specified in the initialization parameters BACKGROUND_DUMP_DEST and USER_DUMP_DEST is set up properly for tracing.

Fast Segment Growth

A segment collection is a group of extents that make up a single table, index, temporary or rollback segment. The Oracle Server offers a practical method of space allocation to segments as they are required to grow. Oracle allows a segment to have multiple extents, which the server allocates automatically when they are needed. For any segment that grows continuously, it is important to carefully monitor that segment's growth pattern. Storage values for the database should be chosen to ensure new extents are not frequently allocated.

This event test checks whether any of the segments specified by the Tablespace name, Segment name, and Segment type parameters are allocating extents too quickly. If, for any segment, the number of extents allocated since the last event check is greater than the threshold values specified in the threshold arguments, then a warning or alert is generated.

Parameters
1. Tablespace name filter: Filter of the tablespace names to be monitored, or * for all tablespaces. Tablespaces that are either temporary, read-only or off-line are excluded. Default is *.
2. Segment name filter: Filter of the segments to be monitored, or * for all segments. Default is *.
3. Segment type filter: Filter of the segment types to be monitored, or * for all segment types. Segment types CACHE and DEFERRED ROLLBACK are excluded. Default is *.
4. Alert threshold: Threshold for alert or number of extents. Default is 3 extents.
5. Warning threshold: Threshold for warning or number of extents. Default is 2 extents.

Output

Name of segment growing too quickly.

Tablespace name containing segment.

Number of extents segment grew since last event condition check.

Default Frequency

60 seconds. However, a frequency of 1 day is recommended.

User Action

Consider increasing the value of the segment's NEXT storage parameter value so that extents are allocated less frequently.

Maximum Extents

A segment is a collection of extents that make up a single table, cluster, index, temporary or rollback segment. The MAXEXTENTS segment storage parameter specifies the maximum number of extents that can be allocated to the segment. Once a segment has filled the maximum number of extents, any row insertion will fail with an ORA-01631 error message.

This event test checks whether any of the segments specified by the Tablespace name, Segment name, and the Segment type parameters are approaching their maximum extents. If for any segment the maximum number of extents minus the number of existing extents is less than the threshold values specified in the threshold arguments, then a warning or alert is generated.

Example

If the maximum number of extents for a segment is 20 and the number of existing extents is 16, then 4 is compared against the specified threshold values. If 3 is specified for an alert and 5 is specified for a warning, the warning event is triggered because only 4 extents are available.

Parameters

1. Tablespace name filter: Filter of the tablespace names to be monitored, or * for all tablespaces. Tablespaces that are either temporary, read-only, or off-line are excluded. Default is *.
2. Segment name filter: Filter of the segments to be monitored, or * for all segments. Default is *.

3. Segment type filter: Filter of the segment types to be monitored, or * for all segment types. Segment types CACHE and DEFERRED ROLLBACK are excluded. Default is *.
4. Alert threshold: Threshold for alert (number of extents). Default is 1 extent.
5. Warning threshold: Threshold for warning (number of extents). Default is 2 extents.

Output

Name of segment approaching its maximum extents.

Name of tablespace containing segment.

Number of extents that can be allocated before the maximum number of extents is hit.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

If possible, increase the value of the segment's MAXEXTENTS storage parameter. Otherwise, rebuild the segment with a larger extent size ensuring the extents within a segment are the same size by specifying STORAGE parameters where NEXT=INITIAL and PCTINCREASE=0. For segments that are linearly scanned, choose an extent size that is a multiple of the number of blocks read during each multiblock read. This will ensure that the Oracle multiblock read capability is used efficiently.

Multiple Extents

A segment is a collection of extents that make up a single table, cluster, index, temporary or rollback segment. The Oracle Server allows a segment to have multiple extents, which the server allocates automatically when additional space is required.

There is no performance degradation for a segment having multiple extents that are never fully-scanned (table and temporary segments only) where the extents are the same size and are also an integral multiple of the multiblock read batch size. No performance degradation is found where extents are 100 or more times larger than the read batch size. Oracle administrators may, however, choose to monitor the number of extents in a segment.

This event test checks whether any of the segments specified by the Tablespace name, Segment name, and Segment type parameters have multiple extents. If the

number of extents is greater than the threshold values specified in the threshold arguments, then a warning or alert is generated.

> **Note:** The only time multiple extents may cause a performance problem is when a segment is fully scanned and that segment's extent size is *not* a multiple of the multiblock read size.

Parameters

1. Tablespace name filter: Filter of the tablespace names to be monitored, or * for all tablespaces. Tablespaces that are either temporary, read-only, or off-line are excluded. Default is *.
2. Segment name filter: Filter of the segment names to be monitored, or * for all segments. Default is *.
3. Segment type filter: Filter of the segment types to be monitored, or * for all segment types. Segment types CACHE and DEFERRED ROLLBACK are excluded. Default is *.
4. Alert threshold: Threshold for alert (number of extents). Default is 50 extents.
5. Warning threshold: Threshold for warning (number of extents). Default is 30 extents.

Output

Name of segment comprised of multiple extents.

Tablespace name containing segment.

Number of extents currently allocated for the segment.

Default Frequency

60 seconds. However, a frequency of 10 minutes is recommended.

User Action

If the segment may be linearly scanned, make sure the multiple extents are the same size. The chosen extent size is an integral multiple of the multiblock read batch size or the extent size is 100 or more times larger than the read batch size in order to achieve the highest efficiency of the server's multiblock read capability.

For all other segments, no action is required unless the number of extent allocations is approaching the segment's maximum number of extents. In this case, increase the value of the segment's MAXEXTENTS storage parameter if possible.

Otherwise, rebuild the segment with a larger extent size ensuring that the extents within a segment are the same size by specifying STORAGE parameters where NEXT=INITIAL and PCTINCREASE=0.

SnapShot Log Full

A master table's snapshot log keeps track of fast refresh data for all corresponding snapshots. When a snapshot log is created for a master table, Oracle creates an underlying table to support the snapshot log. Oracle automatically tracks which rows in a snapshot log have been used during the refreshes of snapshots, and purges those rows from the log. Oracle does not delete rows from the log until all snapshots have used them. As a result, in certain situations, a snapshot log can grow indefinitely when multiple snapshots are based on the same master table. It is best to always try to keep a snapshot log as small as possible to minimize the database space that it uses.

This event test checks whether a snapshot log is too large. In order to do this, the test determines the number of snapshot log tables containing more rows than specified by the snapshot log's table size parameter. If this number is greater than the threshold value specified in the threshold argument, then an alert is generated.

Parameters

1. Threshold for alert (number of snapshot log tables). Default is 1 snapshot log table.
2. Snapshot log table size (number of rows). Default is 100 rows.

Output

Snapshot log table and its size.

Default Frequency

60 seconds

User Action

To remove rows from a snapshot log and free up space for newer log records, you can refresh the snapshots associated with the log so that Oracle can purge rows from the snapshot log or manually purge records in the log by deleting the rows required only by the *n*th least recently refreshed snapshots.

To manually purge rows from a snapshot log, execute the PURGE_LOG stored procedure of the DBMS_SNAPSHOT package at the database that contains the log.

Tablespace Full

As segments within a tablespace grow, the free space within that tablespace decreases. Should free space become insufficient, the creation of new segments or the extension of existing segments will fail.

This event test checks for the total free space in the tablespace specified by the Tablespace name. If the percentage of used space is greater than the values specified in the threshold arguments, then a warning or alert is generated.

Parameters

1. Tablespace name filter: Filter of the tablespace names to be monitored, or * for all tablespaces. Tablespaces that are either read-only or off-line are excluded. The filter must include SQL syntax, for example, = 'ABC', in ('XYZ', 'ABC'), like '%ABC'. For Intelligent Agent's released before 8.1.6, tablespaces that have at least one datafile with autoextension enabled are excluded. Default is *.
2. Alert threshold: Threshold value for alert (%). Default is 90%.
3. Warning threshold: Threshold value for warning (%). Default is 80%.

Output

- Tablespace name
- Current used space in percentage
- Current free space in kilobytes

Default Frequency

60 seconds

User Action

Increase the size of the tablespace by enabling automatic extension for one of its datafiles, manually resizing one of its datafiles, or adding a new datafile.

Database Resource Management Event Tests

This category of event tests tracks possible resource problems within the database environment.

Datafile Limit

The DB_FILES initialization parameter specifies the maximum number of database files that can be opened for this database.

This event test checks for the utilization of the datafile resource against the values (percentages) specified by the threshold arguments. If the percentage of datafiles currently used to the limit set in the DB_FILES initialization parameter exceeds the values specified in the threshold arguments, then a warning or alert is generated.

Example

If 30 datafiles are used and the value of DB_FILES is 40, the percentage is 75% (30 / 40 x 100). This value is compared against the specified thresholds.

Parameters

1. Alert threshold: Threshold value for alert (%). Default is 90%.
2. Warning threshold: Threshold value for warning (%). Default is 80%.

Output

Current value and the limit specified by DB_FILES.

User Action

Verify the current number of data files in use by the database. Increase the DB_ FILES instance parameter, if the current value for DB_FILES is less than MAXDATAFILES.

Lock Limit

The DML_LOCKS initialization parameter specifies the maximum number of DML locks. The purpose of DML locks is to guarantee the integrity of data being accessed concurrently by multiple users. DML locks prevent destructive interference of simultaneous conflicting DML and / or DDL operations.

This event test checks for the utilization of the lock resource against the values (percentage) specified by the threshold arguments. If the percentage of all active DML locks to the limit set in the DML_LOCKS initialization parameter exceeds the values specified in the threshold arguments, then a warning or alert is generated.

If DML_LOCKS is 0, this event fails to register. A value of 0 indicates that enqueues are disabled.

Example

If 40 DML locks are active and the value of DML_LOCKS is 60, the percentage is 67% (40 / 60 x 100). This value is compared against the specified thresholds.

Parameters

1. Alert threshold: Threshold value for alert (%). Default is 90%.

2. Warning threshold: Threshold value for warning (%). Default is 80%.

Output
Current value and the limit specified by DML_LOCKS.

Default Frequency
60 seconds

User Action
Increase the DML_LOCKS instance parameter by 10%.

Process Limit

The PROCESSES initialization parameter specifies the maximum number of operating system user processes that can simultaneously connect to a database at the same time. This number also includes background processes utilized by the instance.

This event test checks for the utilization of the process resource against the values (percentage) specified by the threshold arguments. If the percentage of all current processes to the limit set in the PROCESSES initialization parameter exceeds the values specified in the threshold arguments, then a warning or alert is generated.

Example
If 40 processes are currently connected and the value of PROCESSES is 50, the percentage is 80% (40 / 50 x 100). This value is compared against the specified thresholds.

Parameters
1. Alert threshold: Threshold value for alert (%). Default is 90%.
2. Warning threshold: Threshold value for warning (%). Default is 80%.

Output
Current value and the limit specified by PROCESSES.

Default Frequency
60 seconds

User Action
Verify that the current PROCESSES instance parameter setting has not exceeded the operating system-dependent maximum. Increase the number of processes to be at least 6 + the maximum number of concurrent users expected to log in to the instance.

Session Limit

The SESSIONS initialization parameter specifies the maximum number of concurrent connections that the database will allow.

This event test checks for the utilization of the session resource against the values (percentage) specified by the threshold arguments. If the percentage of the number of sessions, including background processes, to the limit set in the SESSIONS initialization parameter exceeds the values specified in the threshold arguments, then a warning or alert is generated.

Example

If there are 20 sessions and the value of SESSIONS is 25, the percentage is 80% (20 / 25 x 100). This value is compared against the specified thresholds.

Parameters

1. Alert threshold: Threshold value for alert (%). Default is 90%.
2. Warning threshold: Threshold value for warning (%). Default is 80%.

Output

Current value and the limit specified by SESSIONS.

Default Frequency

60 seconds

User Action

Increase the SESSIONS instance parameter. For XA (transaction application) environments, confirm that SESSIONS is at least 2.73 * PROCESSES. For MTS (multi-thread server) environments, confirm that SESSIONS is at least 1.1 * maximum number of connections.

User Limit

The LICENSE_MAX_SESSIONS initialization parameter specifies the maximum number of concurrent user sessions allowed simultaneously.

This event test checks whether the number of users logged on is reaching the license limit. If the percentage of the number of concurrent user sessions to the limit set in the LICENSE_MAX_SESSIONS initialization parameter exceeds the values specified in the threshold arguments, then a warning or alert is generated. If LICENSE_MAX_SESSIONS is not explicitly set to a value, the event does not trigger.

Note: This event test is most useful when session licensing is enabled. Refer to the *Oracle Server Reference Manual* for more information on LICENSE_MAX_SESSIONS and LICENSE_MAX_USERS.

Example

If there are 15 concurrent user sessions and the value of LICENSE_MAX_SESSIONS is 20, the percentage is 75% (15/20 x 100). This value is compared against the specified thresholds.

Parameters

1. Alert threshold: Threshold value for alert (%). Default is 90%.
2. Warning threshold: Threshold value for warning (%). Default is 80%.

Output

Current value and the limit specified by SESSIONS.

Default Frequency

60 seconds

User Action

This typically indicates that the license limit for the database has been reached. The user will need to acquire additional licenses, then increase LICENSE_MAX_SESSIONS to reflect the new value.

Database Performance Management Event Tests

This category of event tests monitors the system for performance problems.

Buffer Cache

The data block buffer cache efficiency, as measured by the hit ratio, records the percentage of times the data block requested by the query is in memory.

Effective use of the buffer cache can greatly reduce the I/O load on the database. If the buffer cache is too small, frequently accessed data will be flushed from the buffer cache too quickly which forces the information to be re-fetched from disk. Since disk access is much slower than memory access, application performance will suffer. In addition, the extra burden imposed on the I/O subsystem could introduce a bottleneck at one or more devices which would further degrade performance.

This event test monitors the buffer cache hit ratio (percentage of success) against the values specified by the threshold arguments. If the number of occurrences is smaller than the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that cache hit ratio is below thresholds before an alert or warning is generated.
2. Alert threshold: Threshold for alert (%). Default is 70%.
3. Warning threshold: Threshold for warning (%). Default is 80%.

Output

Current ratio.

Default Frequency

60 seconds

User Action

The DB_BLOCK_BUFFERS initialization parameter determines the number of database buffers available in the buffer cache. It is one of the primary parameters which contribute to the total memory requirements of the SGA on the instance. The DB_BLOCK_BUFFERS parameter, together with the DB_BLOCK_SIZE parameter, controls the total size of the buffer cache. Since DB_BLOCK_SIZE can only be specified when the database is first created, normally the size of the buffer cache size is controlled using the DB_BLOCK_BUFFERS parameter.

Consider increasing the DB_BLOCK_BUFFERS initialization parameter in order to increase the size of the buffer cache. This increase allows the Oracle Server to keep more information in memory, thus reducing the number of I/O operations required to do an equivalent amount of work using the current cache size.

Continued Row

In two circumstances the data for a row in a table may be too large to fit into a single data block. This results in row fragmentation.

In the first case, the row is too large to fit into one data block when it is first inserted. In this case, the Oracle Server stores the data for the row in a chain of data blocks reserved for that segment. Row chaining (or continuation) most often occurs with large rows, such as rows that contain a column of datatype LONG or LONG RAW. Row continuation in these cases is unavoidable without using a larger DB_BLOCK_SIZE.

In the second case, however, a row that originally fit into one data block is updated so that the overall row length increases and the block's free space is already completely filled. In this case, Oracle migrates the data for the entire row to a new data block, assuming the entire row can fit into a new block. Oracle preserves the original row piece of a migrated row to point to the new block containing the migrated row.

When a row is continued or migrated, I/O performance associated with this row decreases because Oracle must scan more than one data block to retrieve the information for the row.

This event test monitors whether continued rows are found in the segments specified by the Segment name, Segment owner, and Segment type parameters. If continued rows are found, an alert is generated.

Parameters

1. Segment name filter: Filter of the segment names to be monitored, or * for all segments. Default is *.
2. Segment owner filter: Filter of the segment owners to be monitored, or * for all owners. Default is *. Owners SYS and SYSTEM are excluded.
3. Segment type filter: Filter of the segment types to be monitored, or * for all segment types. Default is *. Only segment types TABLE PARTITION and CLUSTER are allowed.

Output

Names of segments containing continued or migrated rows.

Default Frequency

60 seconds. However, a frequency of 1 day is recommended.

Note: This test is CPU-intensive. You may want to schedule the test for once a day at non-business hours.

User Action

If a segment containing fragmented rows has been detected, there are two ways to solve the problem. If rows are not likely to continue growing, rebuild the table. Row fragmentation is eliminated as rows are tightly packed into each database block during re-creation.

If rows are likely to continue growing through updates, consider increasing the segment's PCTFREE value to reduce the likelihood of future row fragmentation.

Data Dictionary Cache

The shared pool is an area in the SGA that contains the library cache of shared SQL requests, the dictionary cache and the other cache structures that are specific to a particular instance configuration.

The dictionary cache efficiency, as measured by the miss ratio, records the percentage of times the dictionary data was not already in memory.

The shared pool mechanism can greatly reduce system resource consumption in at least three ways:

1. Parse time is avoided if the SQL statement is already in the shared pool.
2. Application memory overhead is reduced, since all applications utilize the same pool of shared SQL statements and dictionary resources.
3. I/O resources are saved, since dictionary elements which are in the shared pool do not require access.

If the shared pool is too small, users will consume additional resources to complete a database operation. For dictionary cache access, the overhead is primarily the additional I/O since the dictionary cache references that have been displaced from the cache will need to be re-fetched from disk.

This event test monitors the data dictionary cache miss ratio (percentage of failures) against the values specified by the threshold arguments. If the number of occurrences exceeds the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that cache miss ratio exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (%). Default is 15%.
3. Warning threshold: Threshold for warning (%). Default is 10%.

Output

Current ratio.

Default Frequency

60 seconds

User Action

The SHARED_POOL_SIZE initialization parameters controls the total size of the shared pool. Consider increasing SHARED_POOL_SIZE in order to decrease the

frequency in which dictionary data is being flushed from the shared pool in order to make room for new data.

Disk I/O

This event test monitors the real time database physical I/O rate (requests/seconds) against the values specified by the threshold arguments. If the disk I/O rate exceeds the threshold values entered for the specified number of occurrences, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that I/O rate exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (requests/seconds). No default provided.
3. Warning threshold: Threshold for warning (requests/seconds). No default provided.

Output

Current rate in requests/second.

Default Frequency

60 seconds

User Action

Determine whether the I/O rate is having a negative impact to performance by investigating the disk queue lengths for high I/O devices. It may be necessary to move data files around to balance any identified bottlenecks. Other tuning efforts such as adjusting indexes to reduce the number of full table scans can also reduce I/O load.

If no bottlenecks are evident, increase the I/O rate threshold values.

Free Buffer

Database writer process (DBWR) bottlenecks can be detected by monitoring occurrences of the free buffer waits test over time. If the database environment is in a steady state, there should not be any free buffer waits. However, an occasional absolute increase in free buffer waits is not a problem. Only consistent occurrences of an increase should be of concern.

As a result, this test maintains a history of free buffer waits samples, specified by the Number of Samples parameter, and monitors for a percentage of these samples

where an increase was detected. This percentage is then compared against the values specified by the threshold arguments. If the percentage of samples (where an increase in free buffer waits is detected) is larger than the threshold arguments, then an alert or warning is generated.

Example: If 10 has been specified for the number of samples, then during the first 9 times the test condition is checked, the test is merely building up the history of free buffer waits samples. On the 10 interval and from that point on, the test monitors how many of those samples showed and increase in free buffer waits. Assume 2 samples showed an increase, then the percentage of samples showing an increase is 20%.

Parameters

1. Number of samples: Number of free buffer waits samples. Default is 10.
2. Alert threshold: Threshold for alert (%). Default is 20%.
3. Warning threshold: Threshold for warning (%). Default is 10%.

Output

- Current percentage of samples where an increase in free buffer waits is detected.
- Current DB_FILE_SIMULTANEOUS_WRITES setting.

Default Frequency

60 seconds

User Action

When users are having to wait for free buffers, then either DB_FILE_ SIMULTANEOUS_WRITES needs to be increased or the number of DBWR processes needs to be increased.

The DB_FILE_SIMULTANEOUS_WRITES initialization parameter determines the number of simultaneous writes to each database file when written by DBWR. This parameter is also used to determine the number of reads per file in the redo read ahead when reading redo during recover. This parameter impacts the number of simultaneous I/Os, not just the number of simultaneous writes.

Consider increasing the DB_FILE_SIMULTANEOUS_WRITES initialization parameter in order to increase the speed at which the DBWR writes dirty buffers which then decreases the number of times sessions needed to wait for free buffers.

The DB_WRITES initialization parameter controls the number of DBWR processes that are activated at instance startup. It is a platform specific parameter which is

used to avoid DBWR bottlenecks on operating systems which do not support asynchronous I/O. The DBWR process is responsible for writing dirty buffers in batches from the buffer cache back to the datafiles.

DBWR bottlenecks are most likely on systems which have a high insert, update or delete rate and a large number of disk devices. Since database writes are not serial, there can be benefit to having multiple DBWR processes, even in a single CPU database environment.

Index Rebuild

When an indexed value is updated in the table, the old value is deleted from the index and the new value is inserted into a separate part of the index. The space released by the old value can never be used again. As indexed values are updated or deleted, the amount of unusable space within the index increases, a condition called index stagnation. Because a stagnated index contains a mixture of data and empty areas, scans of the index will be less efficient.

This event test monitors whether indexes specified by the Index name, Index owner, Indexed object name, and Indexed object owner name parameters suffer from index stagnation. If an index has stagnation, an alert is generated.

Parameters

1. Index name filter: Filter of the index names to be monitored, or * for all indexes. Default is *.
2. Index owner filter: Filter of the index owners to be monitored, or * for all owners. Owners SYS and SYSTEM are excluded. Default is *.
3. Index object name filter: Filter of the indexed object name to be monitored, or * for all objects. Default is *.
4. Indexed object owner name filter: Filter of the indexed object owner name to be monitored, or * for all owners. Default is *.

Output

Name of index where index stagnation is detected.

Default Frequency

60 seconds. However, a frequency of 1 day is recommended.

User Action

Consider rebuilding the index to enhance performance. An index rebuild can be accomplished by using either the ALTER INDEX REBUILD statement or the CREATE INDEX statement.

In Memory Sorts

The sort efficiency is measured by the percentage of times sorts were performed in memory as opposed to going to disk.

For best performance, most sorts should occur in memory as sorts to disks are expensive to perform. If the sort area is too small, extra sort runs will be required during the sort operation. This increases CPU and I/O resource consumption.

This event test monitors the in memory sort ratio. The ratio equals the number of sorts performed in memory divided by the total number of sorts performed. If the number of occurrences is smaller that the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that in memory sort ratio is below the thresholds before an alert is generated. Default is 3.
2. Alert threshold: Threshold for alert (%). Default is 90%.
3. Warning threshold: Threshold for warning (%). Default is 100%.

Output

- Current percentage of sorts being performed in memory.
- Current SORT_AREA_SIZE setting in kilobytes.
- Total SGA size in kilobytes.

Default Frequency

60 seconds

User Action

Consider increasing the SORT_AREA_SIZE initialization parameter in order to increase the size of the sort area which will allow the Oracle Server to keep sorts in memory, reducing the number of I/O operations required to do an equivalent amount of work using the current sort area size.

Library Cache

The shared pool is an area in the SGA that contains the library cache of shared SQL requests, the dictionary cache and the other cache structures that are specific to a particular instance configuration.

The library cache efficiency, as measured by the miss ratio, records the percentage of times the fully parsed or compiled representation of PL/SQL blocks and SQL statements are not already in memory.

The shared pool mechanism can greatly reduce system resource consumption in at least three ways:

1. Parse time is avoided if the SQL statement is already in the shared pool.
2. Application memory overhead is reduced, since all applications utilize the same pool of shared SQL statements and dictionary resources.
3. I/O resources are saved, since dictionary elements which are in the shared pool do not require access.

If the shared pool is too small, users will consume additional resources to complete a database operation. For library cache access, the overhead is primarily the additional CPU resources required to re-parse the SQL statement.

This event test monitors the library cache miss ratio (percentage of failures) against the values specified by the threshold arguments. If the number of occurrences exceeds the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that cache hit ratio exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (%). Default is 1%.
3. Warning threshold: Threshold for warning (%). Default is 0.5%.

Output

Current ratio.

Default Frequency

60 seconds

User Action

The SHARED_POOL_SIZE initialization parameter controls the total size of the shared pool. Consider increasing the SHARED_POOL_SIZE in order to decrease the

frequency in which SQL requests are being flushed from the shared pool in order to make room for new requests.

To take advantage of the additional memory available for shared SQL areas, you may also need to increase the number of cursors permitted per session. You can increase this limit by increasing the value of the initialization parameter OPEN_CURSORS.

Also examine SQL statements which can be modified to optimize shared SQL pool memory use and avoid unnecessary statement reparsing. This type of problem is commonly caused when similar SQL statements are written which differ in space, case, or some combination of the two. You may also consider using bind variables rather than explicitly specified constants in your statements whenever possible.

Net I/O

This event test monitors the network I/O rate (bytes/seconds) against the values specified by the threshold arguments. If the network I/O rate exceeds the threshold values entered for the specified number of occurrences, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that I/O rate exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (bytes/seconds). No default provided.
3. Warning threshold: Threshold for warning (bytes/seconds). No default provided.

Output

Current rate in bytes/seconds.

Default Frequency

60 seconds

User Action

Investigate whether the current I/O rate condition is having a negative performance impact on the applications. If it is determined that the excessive network traffic is a performance problem, several approaches can be used to eliminate the bottleneck. Either the network bandwidth can be increased by introducing faster hardware, or applications can be redesigned to take advantage of features such as array fetch, which minimize network traffic.

Redo Log Allocation

Redo log entries contain a record of changes that have been made to the database block buffers. The log writer (LGWR) process writes redo log entries from the log buffer to a redo log file. The log buffer should be sized so that space is available in the log buffer for new entries, even when access to the redo log is heavy. When the log buffer is undersized, user process will be delayed as they wait for the LGWR to free space in the redo log buffer.

The redo log buffer efficiency, as measured by the hit ratio, records the percentage of times users did not have to wait for the log writer to free space in the redo log buffer.

This event test monitors the redo log buffer hit ratio (percentage of success) against the values specified by the threshold arguments. If the number of occurrences is smaller than the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that cache hit ratio is below the thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (%). Default is 98%.
3. Warning threshold: Threshold of warning (%). Default is 100%.

Output

- Current ratio.
- Current LOG_BUFFER in kilobytes.

Default Frequency

60 seconds

User Action

The LOG_BUFFER initialization parameter determines the amount of memory that is used when buffering redo entries to the redo log file.

Consider increasing the LOG_BUFFER initialization parameter in order to increase the size of the redo log buffer. Redo log entries contain a record of the changes that have been made to the database block buffers. The log writer process (LGWR) writes redo log entries from the log buffer to a redo log. The redo log buffer should be sized so space is available in the log buffer for new entries, even when access to the redo log is heavy.

Rollback Contention

Rollback segments are portions of the database that record the actions of transactions in case a transaction is rolled back. Rollback segments are used to provide read consistency, support rollback transactions and recover a database.

Proper allocation of rollback segments make for optimal database performance. Using a sufficient number of rollback segments distributes rollback segment contention across many segments and improves performance.

Contention for rollback segments is reflected by contention for buffers that contain rollback segment blocks.

This event test monitors rollback segment hit ratio (percentage) against the values specified by the threshold arguments. If the rollback segment hit ratio is greater than the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences that rollback segment hit ratio is below the thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (%). Default is 1%.
3. Warning threshold: Threshold for warning (%). Default is 0%.

Output

- Current ratio.
- Current number of concurrently active transactions.
- Current number of online rollback segments.

Default Frequency

60 seconds

User Action

To reduce contention for buffers containing rollback segment blocks, create additional rollback segments. The general guidelines for choosing how many rollback segments to allocate is based on the number and type of concurrently active transactions on your database. Allocate one rollback segment for each 10 concurrent OLTP (online transaction processing) transactions and one rollback segment for each concurrent batch job.

In addition, when creating a rollback segment, keep extents within a rollback the same size by specifying STORAGE parameters where NEXT=INITIAL.

It is also recommended that you set the MINEXTENTS parameter value to 20. Because rollback segments are logically regarded as a circular queue of extents, they are required to have MINEXTENTS value of at least 2. The probability that a rollback segment will require a new extent allocation depends on how likely the next extents are to contain active undo. The more extents the rollback segment has, the less likely that a rollback segment will require an extent allocation that could be avoided. Administrators should create rollback segments with many extents. Naturally, there is a point of diminishing returns. There is a rapid decline of a rollback segment's probability of extending as the number of extents increases. It has been determined that beyond 20 extents, the incremental decrease in the segment's probability of extending fails to justify the cost of the additional extent.

SysStat Table

You can monitor any system statistic available in the database with this test. An alert or warning will be generated if the value of the selected V$SYSSTAT parameter exceeds the values specified by the threshold arguments.

To view the V$SYSSTAT parameter names and values, connect to the database with SQL Worksheet and execute SELECT NAME, VALUE FROM V$SYSSTAT.

Parameters

1. Number of occurrences: Number of consecutive occurrences that exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (parameter value). Default is 15.
3. Warning threshold: Threshold for warning (parameter value). Default is 10.
4. Parameter name: The name of the parameter in the V$SYSSTAT table that you want to monitor. Do not use quotes.

Output

Parameter name and current value.

Default Frequency

60 seconds

User Action

The user action for the event is dependent on the statistic that is being monitored.

SysStat Table Delta

You can monitor any system statistic available in the database with this test. The threshold values are compared to the difference between the last sample point and the current sample point of the V$SYSSTAT parameter. An alert or warning is generated if the calculated difference exceeds the values specified by the threshold arguments.

To view the V$SYSSTAT parameter names and values, connect to the database with SQL Worksheet and execute SELECT NAME, VALUE FROM V$SYSSTAT.

Parameters

1. Number of occurrences: Number of consecutive occurrences that exceeds thresholds before an alert or warning is generated. Default is 3.
2. Alert threshold: Threshold for alert (change in parameter value). Default is 15.
3. Warning threshold: Threshold for warning (change in parameter value). Default is 10.
4. Parameter: The name of the parameter in the V$SYSSTAT table that you want to monitor. Do not use quotes.

Output

Parameter name and change in parameter's value.

Default Frequency

60 seconds

User Action

The user for the event is dependent upon the statistic that is being monitored.

Database Audit Management Event Test

This category of event test allows you to monitor specific database user connections.

User Audit

This event test monitors specified database user connections. For example, an alert is displayed when a particular database user connection, specified by the User name filter argument, has been detected.

Note: For Intelligent Agent's released before 8.1.6, this event test condition never changes from *alert* to *event cleared* even after the user no longer has a current database connection.

Parameters

User Name filter: Filter the user names to be monitored, or * for all users. Default is ='SYS'.

Output

- User name.
- Number of current sessions for user.
- Operating system machine name where user is logged on.

Default Frequency

60 seconds. However, a frequency of 5 minutes is recommended.

User Action

User actions may vary depending on the user connection detected.

Node Fault Management Event Tests

This category of event tests monitors for severe conditions on the system. Immediate action needs to be taken by the administrator.

Data Gatherer Alert Event

This event test signifies that the Intelligent Agent data gathering service has generated errors to the Data Gatherer alert file since the last sample time. The Data Gatherer alert file is a special trace file containing a chronological log of messages and errors. Note that the *Data Gatherer* alert log file is different than the *Database* alert log file. An alert is displayed when Data Gatherer (ODG-xxxxx) messages are written to the Data Gatherer alert file.

Parameters

None

Output

Alert log error messages since last sample time.

Default Frequency
60 seconds

User Action
Examine Data Gatherer alert log for additional information.

Data Gatherer UpDown (Node)

This event test checks whether the Intelligent Agent data gathering service being monitored is running. If the Intelligent Agent data gathering service is down, this event is triggered.

Parameters
None

Output
None

Default Frequency
60 seconds

User Action
Restart the Intelligent Agent data gathering service.

Node UpDown

This event test checks whether the agent on a node can be accessed from the Console. If the Intelligent Agent is down, this test is triggered.

Parameters
None

Output
None

Default Frequency
60 seconds

User Action
Restart the Intelligent Agent.

Node Space Management Event Tests

This category of event tests tracks possible space problems.

Disk Full

This event test checks for available space on the disk specified by the Disk name parameter, such as c: (Windows) or /tmp (UNIX). If the space available is less than the values specified by the thresholds, then a warning or alert is generated.

Parameters

1. Disk name: Name of the disk to be monitored. No default is provided.

> **Note:** "*****" is not a supported disk name.

2. Alert threshold: Free space threshold in kilobytes for alert. No default is provided.
3. Warning threshold: Free space threshold in kilobytes for warning. No default is provided.

Output

Disk name and space available in kilobytes on the disk.

Disk Full (%)

This event test monitors the same file systems as the Disk Full event test. The Disk Full (%) event test, however, returns the percentage of space remaining on the disk destinations.

Parameters

1. Disk name: Name of the disk to be monitored. No default is provided.

> **Note:** "*****" is not a supported disk name.

2. Alert threshold: Percentage of free space threshold for alert. Default is 30%.
3. Warning threshold: Percentage of free space threshold for warning. Default is 50%.

Output
Disk name and percentage of space available on the disk.

Swap Full

This event test checks for available swap space. If the space available falls below the values specified in the threshold arguments, then a warning or alert is generated.

Parameters
1. Alert threshold: Free space threshold in percentage for alert. No default is provided.
2. Warning threshold: Free space threshold in percentage for warning. No default is provided.

Output
Percentage of available space.

Node Performance Management Event Tests

This category of event tests monitors the system for performance problems.

CPU Utilization

This event test checks for the CPU utilization (percentage used) against the threshold values specified by the threshold arguments. If the number of occurrences exceeds the values specified, then a warning or alert is generated.

Parameters
1. Number of occurrences: Number of consecutive occurrences before an alert or warning is generated.
2. Alert threshold: Threshold for alert (%). No default is provided.
3. Warning threshold: Threshold for alert (%). No default is provided.

Output
Current utilization.

CPU Paging

This event test checks the paging rate (kilobytes/second paged in/out) against the threshold values specified by the threshold arguments. If the number of occurrences exceeds the values specified, then a warning or alert is generated.

Parameters

1. Number of occurrences: Number of consecutive occurrences before an alert is generated. Default is 3.
2. Alert threshold: Alert threshold for alert (kilobytes/second). No default is provided.
3. Warning threshold: Warning threshold for warning. No default is provided.

Output

Current rate.

Listener Event Test

This category of event test monitors for catastrophic conditions on the system. Immediate action needs to be taken by the administrator.

SQLnet UpDown

This event test checks whether the listener on the node being monitored is available.

Parameters

None

User Action

The Startup Listener job task can be set up as a fixit job for automatically correcting the problem. To avoid the fixit job executing when the listener is brought down intentionally, turn off the fixit job option.

Numeric Pager Event IDs

The Event Management System provides paging services which notify an administrator with a pager when an event has occurred. Alphanumeric pagers provide a brief text message identifying the event. Numeric pagers provide the numeric pager event IDs to identify the event.

User-Defined SQL Event Test

The User-Defined SQL test allows you to define your own SQL script that evaluates an event test. You can add SQL-based events to the Event Library for future use and/or editing as you would with any other event. Within the SQL script, you provide a datapoint that is used as a return value for comparison with the Alert and Warning Threshold values.

You specify the operator, alert threshold, warning threshold, number of occurrences, and SQL script. The specified operator is used to compare the return value from the script with the supplied thresholds to determine the event condition. The event is triggered after the specified number of occurrences is met.

Parameters

1. Override Preferred Credential: Check this box if you want to change the user name or password or both.
2. Operator: Select one of the following comparison operators:
   - == (equal)
   - < (less than)
   - > (greater than)
   - <= (less than or equal to)
   - >= (greater than or equal to)
   - != (not equal)
3. Alert Threshold: Depending on the SQL script, type a number or a text string.
4. Warning Threshold: Depending on the SQL script, type a number or a text string.
5. Occurrences Preceding Notifications: Type a numeric value indicating how many times this test must return true before an alert flag is displayed in the Console and before a notification is sent.
6. SQL: Type the SQL script you want to use. You can also cut and paste SQL from an existing script.

Output

Value returned by the SQL script

Default Frequency

60 seconds

User Action

The action depends on the SQL script and hence Oracle cannot make any recommendations.

A

Using Advanced Event Tests Effectively

Oracle Advanced Event Tests are integrated into the Oracle Enterprise Manager Console's Event Management System. They are installed as part of the Diagnostics pack and provide DBAs and database users with additional resources to monitor event conditions in their network environment.

This appendix will demonstrate how the Oracle Event Management System and Advanced Event Tests can help DBAs simplify their day to day administrative tasks, enabling them to perform tasks that can provide even greater benefit to the company.

In our examples, the DBA uses events in three different scenarios to proactively monitor potential problems. Proactive management allows the DBA to take action in order to resolve the issue before the problem occurs, minimize unscheduled downtime, and prevent slow response time.

Each of the scenarios provides a description of the problem, a step-by-step preventative action plan, and a resolution summary. The following management tasks are addressed in the scenarios:

- Operational Management
  - The DBA wants to automate his routine monitoring tasks so he can spend his time addressing performance tuning and participating in system design.
- Space Management
  - The DBA wants to ensure a night time batch load does not fail due to insufficient tablespace free space.
- Performance Tuning
  - The DBA wants to ensure there is no performance degradation once a new SQL intensive application is brought online.

Note: For step-by-step instructions on using the Oracle Enterprise Manager Console and Event Management System, please see the *Oracle Enterprise Manager Administrator's Guide*. This guide also provides information on setting up job system and event requirements.

Operational Management

Problem

The DBA realizes that most of his time is spent performing routine, time-consuming tasks which focus on the operational aspect of keeping various databases up and running. To reduce the labor of system management across numerous managed databases, he uses the Oracle Event Management System. The Event System will allow him to proactively monitor the systems for which he is responsible.

He decides to divide his tasks into three different groups (depending on the frequency at which these tasks should be performed):

- Items that should be monitored on a continuous basis
- Items that should be monitored on a daily basis
- Items that should be monitored on a weekly basis

Action

1. Create an event which will monitor the items that should be monitored continuously by following these steps:

   a. Enter the name of the new event, Continuous, in the General tab of the Create Event page. Choose Database as the Destination Type. Accept the Frequency default. Add one or more destinations where the monitoring should take place for the event.

   b. Select the following event tests from the Available Tests column in the Tests tab of the Create Event page:

      Fault:

      Alert-- new errors in Alert file

      Database UpDown-- database is down

Probe -- database connection cannot be made

User Blocks -- database user is blocking others

Performance:

Buffer Cache -- buffer cache hit ratio is low

Free Buffer -- free buffer waits is increasing

In Memory Sorts -- in memory sort ratio is low

Redo Log Allocation -- redo log allocation hit ratio is low

Rollback Contention -- rollback segment contention is high

Space:

Archive Full -- archive device is full

Archive Full (%) -- archive device is full shown as a percentage

c. In the Parameters tab of the Create Event page, accept the Alert and Warning threshold settings.

2. Create an event which will monitor the items that should be monitored on a daily basis by following these steps:

a. Enter the name of the new event, Daily, in the General tab of the Create Event page. Choose Database as the Destination Type. Change the Frequency to be 24 hours. Add one or more destinations where the monitoring should take place for the event.

b. Select the following event tests from the Available Tests column in the Tests tab of the Create Event page:

Space:

Chunk Small-- contiguous space is insufficient

Dump Full -- dump destination device is full

Dump Full (%)-- dump destination device is full shown as a percentage

Fast Segment Growth-- segment growth rate is high

Maximum Extents -- segment's maxextent limit approached

Tablespace Full -- tablespace is full

c. In the Parameters tab of the Create Event page, accept all default settings.

3. Create an event which will monitor the items that should be monitored on a weekly basis by following these steps:

   a. Enter the name of the new event, Weekly, in the General tab of the Create Event page. Choose Database as the Destination Type. Change the Frequency to be 168 hours (7 days x 24 hours). Add one or more destinations where the monitoring should take place for the event.

   b. Select the following event tests from the Available Tests column in the Tests tab of the Create Event page.

      Resource:

      Datafile Limit -- maximum datafile approached

      Space:

      Alert File Large -- alert file is large

      Performance:

      Index Rebuild-- index which may benefit from a rebuild

   c. In the Parameters tab of the Create Event page, accept all default settings.

Resolution

The appropriate resolution depends on the specific event condition. Refer to "Oracle Advanced Event Tests" on page 6-1 for more information on the events. Each event description provides a possible user action to correct your condition.

In addition, the DBA can use other Oracle Enterprise Manager tools to further pinpoint database inefficiencies and tune performance. Please refer to the Oracle Enterprise Manager documentation library for more information on the individual tools.

Table A–1 Summary of Event Tests Used in Operational Management

Fault	Performance	Space	Resource
Alert	Buffer Cache	Alert File Large	Datafile Limit
Databse UpDown	Free Buffer	Archive Full	
Probe	In Memory Sorts	Archive Full (%)	
User Blocks	Index Rebuilt	Chunk Small	
	Redo Log Allocation	Dump Full	
	Rollback Contention	Dump Full (%)	
		Fast Segment Growth	
		Maximum Extents	
		Tablespace Full	

Space Management

Problem

The DBA has been warned that a new table will be created in the APP_DATA tablespace and during the night a batch load will take place. The DBA decides to monitor the free space in this tablespace. Should free space become a problem, he wants to be notified of the condition, but also wants a datafile automatically added to avoid having the batch load fail.

Action

1. Create a fixit job (a job used to correct problems detected by an event) through the Oracle Enterprise Manager Console Job System by following these steps:
   a. Enter the name of the new job, AppDataAddFile, in the General tab of the Create Job page. Choose Database as the Destination Type. Check the Fixit Job box to indicate this job will be used to correct an event condition. Select the database destination where the event will be monitored.
   b. Select Run SQL*Plus Script from the Available Tasks list in the Tasks tab of the Create Job page.
   c. Enter the following Script Text in the Parameters tab of the Create Job page:

      ```
      ALTER TABLESPACE APP_DATA
      ADD DATAFILE
      ```

'c:\orant\database\app02.ora' SIZE 5M;

2. Submit the fixit job. (You must resubmit the job for each datafile or the job will fail.)
3. Create an event that will monitor the total used space in the APP_DATA tablespace. The event will generate a warning or alarm if the tablespace becomes too full. In addition, associate the AppDataAddFile fixit job, to the event. Then, if the tablespace becomes too full, the job will run and automatically correct the problem. The steps to create the event are:
   a. Enter the name of the new event, AppDataFull, in the General tab of the Create Event page. Choose Database as the Destination Type. Add one or more destinations where the monitoring should take place for the event.
   b. Select the following event test from the Available Tests column in the Tests tab of the Create Event page:

      Space:

      Tablespace Full -- tablespace is full

   c. Select AppDataAddFile from the pull-down list of available fixit jobs in the Parameters tab of the Create Event page. In addition, specify the following filter for the Tablespace Name Filter:

      ='APP_DATA'

   d. In the Parameters tab of the Create Event page, accept the default Warning and Alert threshold settings.

Resolution

In this example using the Oracle Event Management System, the DBA was able to proactively monitor for the potential problem and automatically prevent any impact to end users. No further action would be required.

Performance Tuning

Problem

The DBA knows a new SQL intensive application is being brought on-line. He worries that his shared pool might be sized too small to handle the new application. If the shared pool is too small, users will consume additional resources in order to complete a database operation. For library cache access, the overhead is primarily

the additional CPU resources required to reparse the SQL statement. For dictionary cache access, the overhead is primarily the additional I/O since the dictionary cache references that have been displaced from the cache will need to be refetched from disk.

To prevent any performance degradation that would be noticed by users, he decides to proactively monitor the performance of the shared pool.

Action

Create an event through the Oracle Enterprise Manager Event Management System that will monitor the performance of the shared pool. Since the shared pool contains the library cache and the data dictionary cache, the efficiency of both of these caches should be monitored to determine whether or not the shared pool requires tuning. If either of these caches are performing poorly, the DBA wants to be notified. The steps to create the event set are:

1. Enter the name of the new event, SharedPool, in the General tab of the Create Event page. Choose Database as the Destination Type. Add one or more destinations where the monitoring should take place for the event.
2. Select the following event tests from the Available Tests column in the Tests tab of the Create Event page:

   Performance:

   Library Cache -- monitors library cache efficiency as measured by the miss ratio. It records the percentage of times the fully parsed or compiled representation of PL/SQL blocks and SQL statements are not already in memory.

   Data Dictionary Cache -- monitors dictionary cache efficiency as measured by the miss ratio. It records the percentage of times the dictionary data was not already in memory.
3. Accept the default number of occurrences and threshold settings in the Parameters tab of the Create Event page for each of the selected events.

Resolution

If the library cache miss ratio or the data dictionary miss ratio is high then the DBA has several options:

- Use Oracle Expert to tune the shared pool by creating a focused tuning session for the database. The tuning scope of the tuning session should focus on SGA

instance tuning. If the Library Cache Event test was triggered, the DBA may also want to include the SQL reuse tuning focus. Once Oracle Expert has automatically collected and analyzed the appropriate data, a set of recommendations for achieving the best performance is provided.

- Consider increasing the SHARED_POOL_SIZE to:
  - Decrease the frequency in which SQL requests are being flushed from the shared pool in order to make room for new request
  - Decrease the frequency in which dictionary data is being flushed from the shared pool in order to make room for new data
- To take advantage of the additional memory available for shared SQL areas, the number of cursors permitted for a session may need to be increased. This limit can be increased by increasing the value of the initialization parameter OPEN_CURSORS.
- If the library cache was found to be inefficient, examine SQL statements which can be modified to optimize shared SQL pool memory utilization and avoid unnecessary statement reparsing. This type of problem is commonly caused when similar SQL statements are written which differ in space, case or some combination of the two. Consider using bind variables rather than explicitly specified constants in your statements whenever possible.

Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

Z