# Oracle® Enterprise Manager

Oracle® Trace User's Guide

Release 2.0

January 1999

Part No. A67837-01

Oracle Trace User’s Guide, Release 2.0

Part No. A67837-01

Release 2.0

Primary Author: Jacqueline Gosselin

Contributors: Chelsea Christenson, Leslie Gloyd, George Lavash, Ken Morse, Therese Zimmerman, and other members of the Oracle Trace Engineering Team

# Contents

Send Us Your Comments ........ ix

Preface ........ xi

1 Overview

Oracle Trace Manager ........ 1-1
Managing Collections ........ 1-2
Collecting Event Data ........ 1-3
Accessing Collected Data ........ 1-4
Oracle Trace Environment ........ 1-5
Components on the Client ........ 1-5
Components on the Server ........ 1-6
Checklist of Components Needed to Run Oracle Trace ........ 1-7

2 Using Oracle Trace

Starting Oracle Trace ........ 2-1
Discovering Products ........ 2-2
Results of Discovery ........ 2-5
Viewing the Oracle Trace Manager Main Window ........ 2-5
Left Pane ........ 2-6
Right Pane ........ 2-7
Toolbar Icons ........ 2-7
Viewing Oracle Trace Manager Menus ........ 2-8
File Menu ........ 2-9

Edit Menu .......... 2-9
View Menu .......... 2-9
Collection Menu .......... 2-10
Product Menu .......... 2-11
Help Menu .......... 2-11

## 3 Working with Collections

**Creating Collections** .......... 3-1
Step 1: Select Server Nodes .......... 3-3
Step 2: Choose Oracle Installation .......... 3-3
Step 3: Select Event Sets .......... 3-4
Step 4: Limit Data Collected by Users .......... 3-6
Step 5: Limit Data Collected by Wait Events .......... 3-8
Step 6: Name Collection Definition File .......... 3-10
Step 7: Schedule the Collection .......... 3-12
Advanced Scheduling Options .......... 3-13
Step 8: Select Formatting Option .......... 3-16
Advanced Formatting Options .......... 3-17
Summary Page .......... 3-18
**Viewing Collection Definition and Properties** .......... 3-19
**Creating Like Collections** .......... 3-22
**Stopping Collections** .......... 3-22
**Formatting Collections** .......... 3-22
Preparing to Format Oracle Trace Data .......... 3-23
Formatting a Collection Using Oracle Trace Manager .......... 3-23
**Deleting Collection Entries** .......... 3-24
**Deleting Formatted Data** .......... 3-25
**Deleting Collection Files** .......... 3-25
**Running Reports** .......... 3-26

## 4 Oracle Trace Data Viewer

**Overview** .......... 4-1
**Usage Scenarios for Data Viewer** .......... 4-2
**Starting Data Viewer** .......... 4-2
Starting Data Viewer from Within Oracle Trace .......... 4-3

Starting Data Viewer from Oracle Enterprise Manager........................................................ 4-3
Starting Data Viewer from the Start Menu ................................................................. 4-4
**Using Data Viewer**.................................................................................................... 4-4
Computing Statistics – Preparing Data for Fast Access and Analysis................................ 4-5
Displaying Data Views ................................................................................................ 4-5
Accessing the Definition of a Data View........................................................................ 4-10
Viewing Data in a Data View ........................................................................................ 4-11
Creating Custom Data Views ........................................................................................ 4-13
Modifying or Extending Data Views.............................................................................. 4-13
Drill-Down of Data Views.............................................................................................. 4-14
Selecting a Drill-Down Data View ................................................................................ 4-17
Modifying Drill-Down Data Views ................................................................................ 4-18
Saving As Data Views and Files.................................................................................... 4-19
Printing Data Views ...................................................................................................... 4-19
**Using the Data View Wizard** ...................................................................................... 4-19
Choosing an Event......................................................................................................... 4-20
Choosing Items Associated with the Selected Event......................................................... 4-21
Choosing the Sort Criteria.............................................................................................. 4-23
Defining the Number of Rows to Display ....................................................................... 4-23
Reviewing the Summary Page......................................................................................... 4-24

## A Manual Collection of Oracle Trace Data

**Using the Oracle Trace Command-Line Interface** ....................................................... A-1
**Using Initialization Parameters to Control Oracle Trace** ............................................ A-3
Enabling Oracle Trace Collections for Oracle8 Server ..................................................... A-3
Determining the Event Set that Oracle Trace Collects ..................................................... A-4
**Using Stored Procedure Packages to Control Oracle Trace** ......................................... A-5
**Oracle Trace Collection Results** ................................................................................. A-6
**Formatting Oracle Trace Data to Oracle Tables**.......................................................... A-7
**New otrcfmt Image for Oracle Server Releases 7.3.4 and 8.0.4** ................................... A-8
**Oracle Trace Statistics Reporting Utility**..................................................................... A-9

## B Oracle Server Events

**Data Items Collected for Events**................................................................................. B-2
Resource Utilization Items ............................................................................................ B-2

Cross-Product Items ........ B-3
Items Specific to Oracle Server Events ........ B-5
**Items Associated with Each Event** ........ B-9
Event Statistics Block ........ B-10
Connection Event ........ B-10
Disconnect Event ........ B-11
ErrorStack Event ........ B-11
Migration Event ........ B-12
ApplReg Event ........ B-12
RowSource Event ........ B-12
RowSource Event (Specific to Oracle Server Release 8.0.2 and Higher) ........ B-13
SQLSegment Event ........ B-13
Wait Event ........ B-14
Parse Event ........ B-14
Execute Event ........ B-15
Fetch Event ........ B-15
LogicalTX Event ........ B-16
PhysicalTX Event ........ B-17

## C Troubleshooting Oracle Trace

**Typical Problems Encountered While Using Oracle Trace** ........ C-1
**Server Environment** ........ C-5
**Oracle Enterprise Manager Configuration** ........ C-6
**Network Configuration** ........ C-8
**Oracle Intelligent Agent (UNIX Specific)** ........ C-8
**Oracle Intelligent Agent (NT Specific)** ........ C-9
**Database Configuration** ........ C-9
**Oracle Trace Configuration** ........ C-12
Data Viewer Problems ........ C-13
Could Not View Oracle Trace Formatted Data ........ C-13
Need to Restore Formatted Data ........ C-14
Wait Times Were Not Collected ........ C-14
Additional Formatted Data Is Missing from Data Viewer ........ C-14
Missing SQL Statement from Collection ........ C-14
Collection Is Too Large ........ C-14

Poor Performance .......... C-15
Table or View Does Not Exist (or No Data in Collection) .......... C-15
Oracle Trace Manager Problems .......... C-15
Failure Connecting to Show User List .......... C-16
Format of Collection Failed .......... C-16
Collection Is Empty .......... C-16
Oracle Trace Could Not Access Memory .......... C-17
**Isolating an Oracle Trace Problem** .......... C-17
Isolating a Problem in Oracle Trace Manager .......... C-17
Oracle Enterprise Manager Job System .......... C-18
Formatter Tables .......... C-18
Isolating a Problem in Oracle Trace Collection Services .......... C-19
Oracle Trace Command-Line Interface .......... C-19
Stored Procedures .......... C-22
Oracle Trace Manager Logging .......... C-23
EPC_ERROR.LOG File .......... C-23
Information Needed When Reporting a Problem .......... C-24

## Index

# Send Us Your Comments

**Oracle Trace User's Guide, Release 2.0**

**Part No.  A67837-01**

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 Trace Documentation
- Postal service:  
  Oracle Corporation  
  Oracle Trace Documentation  
  One Oracle Drive  
  Nashua, NH  03062-2698  
  USA

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: *Oracle Enterprise Manager Oracle Trace User's Guide*. Specifically, it covers the following topics:

- *Purpose of This Guide*
- *Audience*
- *How This Guide Is Organized*
- *Conventions Used in This Guide*
- *Oracle Enterprise Manager Pack Information*
- *Oracle Enterprise Manager Documentation*

## Purpose of This Guide

This guide provides user instructions for the Oracle Trace Manager application, which is the Oracle Trace graphical user interface. This guide also provides information on the Oracle Trace Data Viewer used to view formatted data collected by Oracle Trace.

## Audience

This guide is written for Oracle Trace users who want to set up data collections using the Oracle Trace Manager application and who want to view the formatted collected data using Oracle Trace Data Viewer.

## How This Guide Is Organized

This guide is organized as follows:

**Chapter 1, "Overview"**

Provides a general description of Oracle Trace.

**Chapter 2, "Using Oracle Trace"**

Describes how to invoke Oracle Trace, the Oracle Trace Manager windows and menus displayed, and how to exit. Included in this chapter are instructions for discovering products.

**Chapter 3, "Working with Collections"**

Describes how to create, edit, stop, and delete Oracle Trace collections.

**Chapter 4, "Oracle Trace Data Viewer"**

Describes how to use Oracle Trace Data Viewer.

**Appendix A, "Manual Collection of Oracle Trace Data"**

Describes Oracle Trace initialization parameters, stored procedures, and command-line interface.

**Appendix B, "Oracle Server Events"**

Contains information on Oracle Server SQL and wait events and data that can be collected with Oracle Trace.

**Appendix C, "Troubleshooting Oracle Trace"**

Lists the problems you might encounter while using Oracle Trace and provides solutions to these problems.

## Conventions Used in This Guide

The following table lists the conventions used in the guide.

| Convention | Explanation |
|---|---|
| # | The default superuser prompt. |
| % | The default user prompt. |
| [Ctrl/C] | Press the Ctrl key while you simultaneously press another key (in this case, C). |

| Convention | Explanation |
|---|---|
| *italics* | Italicized words indicate variables, such as a file or directory name. |
| $ORACLE_HOME | Represents the directory where you installed Oracle Enterprise Manager components and Oracle Diagnostics Pack products. The directory name may be different on your system. |
| otrace*xx* | The *xx* represents the version of the database for which you are collecting data; for example, otrace73 for the Oracle Server release 7.3. |

This guide also assumes that you are familiar with the operation of Microsoft Windows NT. Refer to the Windows documentation for your system, if necessary. In general, this guide shows the directory names as they are used in UNIX; on an NT system, delineate directory names with a backslash (\).

To reduce wordiness and redundancy, menu and submenu choices are joined by arrows. For example, Collection=>Create refers to the Create Collection choice in the Collection menu.

## Oracle Enterprise Manager Pack Information

The Oracle Enterprise Manager product family includes three packs: Oracle Change Management Pack, Oracle Diagnostics Pack, and Oracle Tuning Pack. Each pack is fully integrated into the Oracle Enterprise Manager Console framework.

**Oracle Change Management Pack**

- Includes DB Alter, DB Capture, DB Diff, DB Propagate, DB Quick Change, DB Search, and Plan Manager.
- 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 safely with easy-to-use wizards that teach the systematic steps necessary to upgrade.

**Oracle Diagnostics Pack**

- Includes Oracle Performance Manager, Oracle Capacity Planner, Oracle Trace, Oracle TopSessions, and Oracle Event Tests that are provided with the Oracle Diagnostics Pack.
- Monitors, diagnoses, and maintains the health of databases, operating systems, and applications. 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 Tuning Pack**

- Includes Oracle Auto-Analyze, Oracle SQL Analyze, Oracle Expert, Oracle Index Tuning Wizard, and the Oracle Tablespace Manager.
- 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.
- Proactively discovers tuning opportunities and automatically generates the analysis and required changes to tune the system. Inherent in the product are powerful teaching tools which train DBAs how to tune as they work.
- Instills "Consultant Quality" tuning expertise into developers and DBAs and increases their productivity.

## Oracle Enterprise Manager Documentation

The *Oracle Enterprise Manager Oracle Trace User's Guide* is one of several Oracle Enterprise Manager documents.

**Oracle Enterprise Manager base documentation**

- The *Oracle Enterprise Manager Readme* provides important notes regarding the online documentation, updates to the software, and other late-breaking information.
- The *Oracle Enterprise Manager Installation* CD-ROM insert provides information about installing Oracle Enterprise Manager.
- The *Oracle Enterprise Manager Administrator's Guide* explains how to use Oracle Enterprise Manager, Oracle's systems management console, common services, and integrated platform tool.

- The *Oracle Enterprise Manager Concepts Guide* provides an overview of the Oracle Enterprise Manager.
- The *Oracle Enterprise Manager Configuration Guide* explains how to configure Oracle Enterprise Manager.
- The *Oracle Enterprise Manager Messages Manual* describes the Oracle Enterprise Manager error messages and methods for diagnosing the messages.

**Oracle Change Management Pack documentation**

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

**Oracle Diagnostics Pack documentation**

- The *Oracle Diagnostics Pack Readme* provides important notes regarding the Diagnostics Pack online documentation, updates to the software, and other late-breaking information.
- The *Oracle Diagnostics Pack Installation* CD-ROM insert provides information about installing the Oracle Diagnostics Pack.
- The *Oracle Enterprise Manager Performance Monitoring and Planning Guide* provides an overview of the concepts and features of the Oracle Performance Manager, Oracle Capacity Planner, and Oracle TopSessions applications. It also describes the Oracle Event Tests that are provided with the Oracle Diagnostics Pack, which allow you to manage database, listener, and service types.
- The *Oracle Enterprise Manager Oracle Trace User's Guide* explains how to use the Oracle Trace application to capture and use historical data to monitor Oracle databases.
- The *Oracle Enterprise Manager Oracle Trace Developer's Guide* explains how to instrument your application with Oracle Trace routines.

**Oracle Tuning Pack documentation**

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

# 1 Overview

Oracle Trace is a general-purpose data collection system that collects data for any software product enabled with Oracle Trace API calls, such as Oracle Server. Use Oracle Trace to collect a wide variety of data, such as performance statistics, diagnostic data, system resource usage, and business transaction details.

The components of Oracle Trace are:

- Oracle Trace Manager
- Oracle Trace Data Viewer
- Oracle Trace Collection Services
- Oracle Trace Application Programming Interface (API)

This manual describes Oracle Trace Manager and Oracle Trace Data Viewer.

Software developers can use the Oracle Trace API to preconfigure, or **instrument,** their own products for Oracle Trace data collection or they can collect data for pre-instrumented products, such as Oracle Server. Users of a product containing the Oracle Trace API calls can then use the Oracle Trace Collection Services to collect data about specific events that occur in that product.

Refer to the *Oracle Enterprise Manager Oracle Trace Developer's Guide* for information about the Oracle Trace Collection Services and how to customize applications using the Oracle Trace API.

## Oracle Trace Manager

Oracle Trace provides a graphical Oracle Trace Manager application to create, schedule, and administer Oracle Trace collections for products containing Oracle Trace API calls.

The Oracle Trace Manager is a client-based Windows application that runs on the Oracle Enterprise Manager console. Oracle Trace Manager automatically discovers Oracle Trace instrumented products that are installed on all nodes that are known to the Oracle Management Server. By default Oracle Trace discovers any Oracle Server release 7.3.3 and higher.

**Note:** Oracle Corporation is currently using the Oracle Trace data collection API in Oracle Server release 7.3 or higher (for performance data collection).

The Oracle Server performance data collected by Oracle Trace includes SQL statements, detailed statistics on SQL events, transaction events, and other useful information.

Oracle Server events and data are described in Appendix B.

## Managing Collections

The use and control of Oracle Trace revolves around the concept of a **collection**. A collection is how you gather data for events that occurred while a product containing Oracle Trace API calls was running.

With Oracle Trace Manager, you can schedule and manage collections. When creating a collection, you define the attributes of the collection, such as the collection name, the products and event sets to be included in the collection, and the start and end time. Oracle Trace Manager includes a Collection Wizard that facilitates the creation and execution of collections.

Once you schedule a collection, it can be executed immediately, scheduled to execute at a specific time, or executed at specified intervals. When a collection executes, it produces a file containing the event data for the products that participated in the collection. You can also use a collection as a template for creating other similar collections.

**Note:** Some instrumented applications also provide their own mechanism for starting and stopping Oracle Trace collections. For example, Oracle Server release 7.3 provides a set of database initialization parameters, described in Appendix A, that you can use to start and stop an Oracle Trace collection. In some situations, using an instrumented product's own controls may be useful. However, you should control most Oracle Trace collections using the Oracle Trace Manager application.

## Collecting Event Data

An **event** is the occurrence of some activity within a product. Part of instrumenting a product with Oracle Trace API calls is deciding which activities to record as events. Oracle Trace only collects data for these predefined events.

There are two types of events:

- Point events

  **Point events** represent an instantaneous occurrence of something in the instrumented product. An example of a point event is an error occurrence.

- Duration events

  **Duration events** have a beginning and ending. An example of a duration event is a transaction. Duration events can have other events occur within them; for example, the occurrence of an error within a transaction.

**Items** are specific pieces of information about an event. If your product has been modified to contain Oracle Trace API calls, the developer has identified specific events and data items. These items, such as a transaction type or dollar amount, are specific to the instrumented product. Data items can also include statistics on the resources used by that event, such as the CPU time and number of input/output operations (I/Os) performed by the event. For example, the Oracle Server release 7.3 and higher has defined 13 events to be collected by Oracle Trace. Three of these events are:

- Database Connection: A point event that records data such as the server login username.
- SQL Parse: One of the series of SQL processing duration events. This event records a large set of data such as SQL cache misses, optimizer mode, user ID, and cursor number.

- RowSource: Data about the execution plan, such as SQL operation, position, object identification, and number of rows processed by a single row source within an execution plan.

A complete list of the server events and data items is contained in Appendix B.

Oracle Trace events can be organized into **event sets** that restrict the data collection to specific events. You can establish event sets for performance monitoring, auditing, diagnostics, or any logical event grouping.

Each event set is described by its own product definition file (.fdf). The **product definition file** is a set of events and their associated data items. The complete set of events defined for an instrumented product is referred to as the ALL event set. Other event sets are then derived from the ALL set. For example, the Oracle Server includes an event set known as the EXPERT set. This set includes SQL event data used by the Oracle Expert tuning application, but excludes other events, such as wait events.

**Note:** Oracle Corporation recommends that you use either the EXPERT or DEFAULT event set when collecting Oracle Server data. The ALL event set generates large amounts of data as it collects wait events. Unless you specifically want to examine wait events, avoid using the ALL event set.

## Accessing Collected Data

During a collection, Oracle Trace buffers event data in memory and periodically writes it to a collection binary file. This method ensures low overhead associated with the collection process. You can access the event data collected in the binary file by formatting the data to an Oracle Server database. This makes the data available for fast, flexible access.

Oracle Trace Manager provides a mechanism for formatting collection data immediately after a collection is run or at a later time.

**Note:** Formatter tables are required in order to format an Oracle Trace collection. This restriction applies to releases of Oracle Server prior to 7.3.4 and 8.0.4. To create Oracle Trace formatter tables, use the vobsh command. See the *Oracle Enterprise Manager Configuration Guide* for more information.

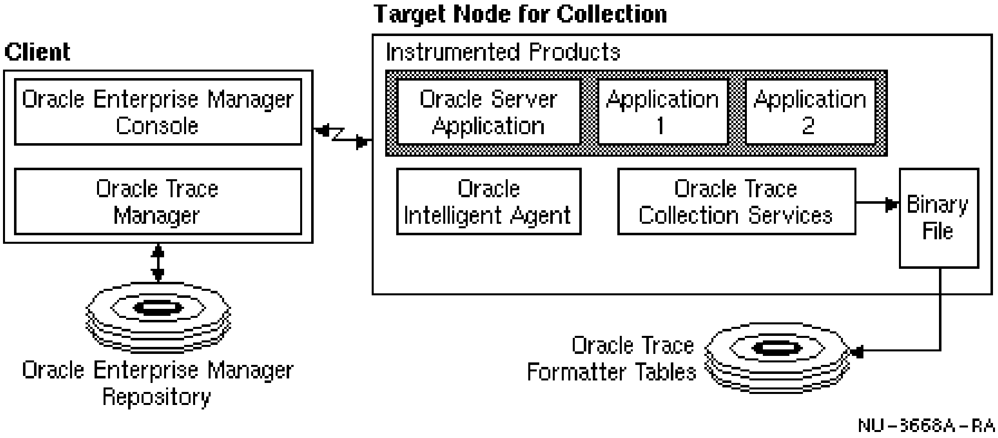
Once the data is formatted, you can access the data using Oracle Trace Data Viewer or by using SQL reporting tools and scripts. Oracle Trace ships with a set of predefined SQL scripts for accessing the formatter tables created for the Oracle Server events. In addition, Oracle Trace data can be preconfigured for use in other applications, for example, for use by Oracle Expert. The Oracle Expert database tuning application can access and analyze the Oracle Server SQL event data from the Oracle Trace formatter tables.

Also, you can access event data by specifying a Detail report from the Oracle Trace reporting utility. This report provides a basic mechanism for viewing a collection's results. You have limited control over what data is reported and how it is presented.

# Oracle Trace Environment

Oracle Trace and its associated components run in a client/server environment. Figure 1–1 shows these components.

***Figure 1–1 Oracle Trace Environment***


Target Node for Collection
Client
Oracle Enterprise Manager Console
Oracle Trace Manager
Instrumented Products
Oracle Server Application
Application 1
Application 2
Oracle Intelligent Agent
Oracle Trace Collection Services
Binary File
Oracle Enterprise Manager Repository
Oracle Trace Formatter Tables
NU–3668A–RA

## Components on the Client

To collect data, the following components must be running on the client:

- Oracle Enterprise Manager console

The Oracle Enterprise Manager repository can reside on the client, on the server, or on an entirely separate machine. Oracle Trace Manager tables reside within the Oracle Enterprise Manager repository. Also, Oracle Trace uses the Oracle Enterprise Manager Job system for scheduling and executing collections.

- Oracle Trace Manager

  The Oracle Trace Manager gets information about managed nodes from the Oracle Management Server. Oracle Trace Manager then communicates information with the Oracle Intelligent Agent running on the managed node to start, stop, format, and delete Oracle Trace collections.

> **Note:** Before you run Oracle Trace for the first time, you must perform discovery within the Oracle Enterprise Manager console. Then, when you run Oracle Trace for the first time, Oracle Trace Manager automatically leads you through the Oracle Trace discovery process of nodes defined within the Oracle Enterprise Manager console.

## Components on the Server

On the server, the following components are required:

- Instrumented products, for example, Oracle Server release 7.3 or higher, and other applications
- Oracle Intelligent Agent

  Oracle Intelligent Agent is an Oracle Enterprise Manager component that ships with the Oracle Server. It contains scripts that control operations submitted by Oracle Trace Manager.
- Oracle Trace Collection Services

  Oracle Trace Collection Services component collects the data and creates the collection definition (.cdf) files and the data collection (.dat) files.

  You can store data collected by Oracle Trace Collection Services in Oracle database tables, referred to as Oracle Trace formatter tables, for access by SQL reporting tools and other products.

> **Note:** Prior to Oracle Server release 8.0.3, Oracle Trace required that stored procedures be installed on the database.
>
> See "Stored Procedures" on page C-22 for additional information.

## Checklist of Components Needed to Run Oracle Trace

This chapter has referred to several Oracle Trace components and other programs that must be configured or running in order for you to use Oracle Trace. This section provides a checklist of these required Oracle Enterprise Manager and Oracle Trace components.

Before you run Oracle Trace, check the following:

- The Oracle Management Server is running, and the nodes that you want to use Oracle Trace on have previously been discovered through the Oracle Enterprise Manager console.
- Oracle Enterprise Manager preferred credentials are set for nodes on which you want to collect data.
- Oracle8 databases have the value of the ORACLE_TRACE_ENABLE parameter in your INIT*sid*.ORA file set to TRUE.
- The Oracle Trace formatter tables have been created (valid for releases of Oracle Server prior to 7.3.4 and 8.0.4).

  These tables are required by the Oracle Trace formatter mechanism, which converts and loads an Oracle Trace collection binary file (*collection_name*.dat) into Oracle tables for access.

  Create these tables in the schema that you want to use for Oracle Trace formatting by using the vobsh command. For additional information about formatter tables, see "Format of Collection Failed" on page C-16.
- Oracle Intelligent Agent is running on the nodes that are targets for Oracle Trace collections.
- For Oracle 7.3 databases, check that the Oracle Trace user account, TRACESVR, and the Oracle Trace stored procedure packages, DBMS_ORACLE_TRACE_AGENT and DBMS_ORACLE_TRACE_USER, exist. If they do not, then 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\otrace*xx*\admin on NT systems.

The otrcsvr.sql script is run automatically during database installation on most platforms. However, if your server platform is NT, you must run this script manually. The Oracle Trace Collection Services control files exist in the $ORACLE_HOME directory on the target nodes.

- The Oracle Trace Collection Services control files exist in the $ORACLE_HOME directory on the target nodes.

  These control files include: regid.dat, process.dat (process.dat has been renamed to facility.dat in Oracle8), and collect.dat. The files are located in $ORACLE_HOME/otrace/admin on UNIX systems and in $ORACLE_HOME\otrace*xx*\admin on Windows NT systems.

  If the control files are missing, run the otrccref.exe executable file, located in the $ORACLE_HOME/bin subdirectory.

- On the Windows NT platform, Oracle Trace Collection Services are made available in the form of a dynamic link library (dll) that must be present at run time. The dynamic link library file is called otrace*xx*.dll, where *xx* represents the release number, and it is located in $ORACLE_HOME\bin.

  On all other platforms, Oracle Trace Collection Services are linked into the application image when the image is built.

# 2

# Using Oracle Trace

Oracle Trace collects data for products that are instrumented with the Oracle Trace data collection API. Use the Oracle Trace Manager application to create and administer Oracle Trace collections for instrumented products.

Oracle Corporation has instrumented Oracle Server release 7.3 and higher with the Oracle Trace data collection API.

As you add additional custom products to nodes that are managed through Oracle Enterprise Manager, you can use Oracle Trace Manager to automatically discover these products and make them available for collection administration.

This chapter includes the following sections:

- *Starting Oracle Trace*
- *Discovering Products*
- *Viewing the Oracle Trace Manager Main Window*
- *Viewing Oracle Trace Manager Menus*

## Starting Oracle Trace

You start Oracle Trace from the Oracle Enterprise Manager console using either the:

- Trace Manager option in the Diagnostics Pack drawer
- Tools=>Diagnostic Pack=>Trace Manager menu option

You can also start Oracle Trace from the Start menu:

Start=>Programs=>ORACLE_HOME=>Diagnostics Pack=>Trace Manager

**Note:** ORACLE_HOME represents the oracle_home in which the Diagnostics Pack is installed.

Provide the login information needed in the Oracle Management Server Login dialog box.

The process of starting an application in Oracle Enterprise Manager is referred to as **launching**.

Because Oracle Trace uses the Oracle Enterprise Manager job subsystem to perform its tasks, the Oracle Management Server must be running at the same time as Oracle Trace.

## Discovering Products

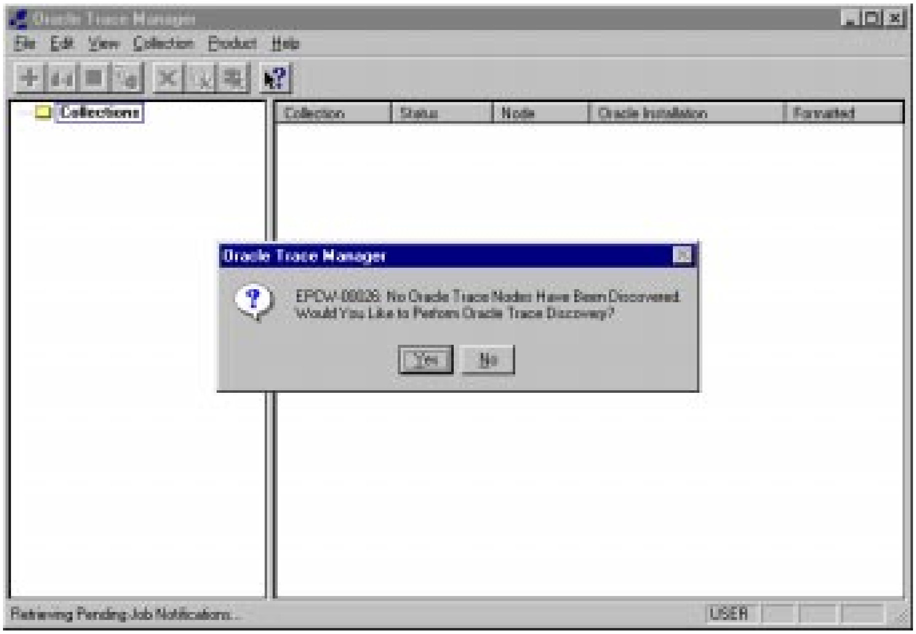
Oracle Trace Manager provides the user with a graphical navigator display of products and database services that are enabled for Oracle Trace data collection. These products and services reside on nodes that are managed by the Oracle Management Server. Products and services enabled with Oracle Trace calls are located or **discovered** by Oracle Trace Manager during the product discovery process.

The Oracle Trace discovery process gathers information on products and services enabled (instrumented) with Oracle Trace calls by submitting batch jobs to specific nodes through the Oracle Enterprise Manager Job system and Oracle Intelligent Agent. These jobs query the managed nodes for products enabled with Oracle Trace API calls and return the results to Oracle Trace Manager, which updates the repository and the Oracle Trace Manager graphical display. The user can only submit Oracle Trace collections for products and services that have been successfully discovered.

**Note:** If you are discovering a node and an error occurs indicating an **authentication error**, you need to set the Oracle Enterprise Manager file preferences for the node to a valid user name and password. See the instructions on setting up the account to run jobs as a user in the *Oracle Enterprise Manager Configuration Guide*.

When you invoke Oracle Trace for the first time, the tree list in the left pane of the Oracle Trace main window is empty; there are no entries under Collections. A message appears asking whether you want to perform an Oracle Trace discovery (see Figure 2–1).

*Figure 2–1 Invoking Oracle Trace for the First Time*

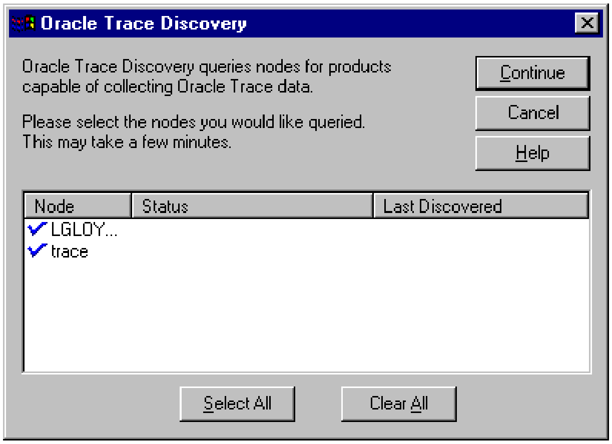


EPCW-00026: No Oracle Trace Nodes Have Been Discovered.
Would You Like to Perform Oracle Trace Discovery?

Retrieving Pending Job Notifications...

To discover products for the first time, click on the Yes button. At any other time, invoke Product=>Discovery. The Oracle Trace Discovery dialog box displays, as shown in Figure 2–2.

*Figure 2–2 Product Discovery Dialog Box*



Oracle Trace Discovery

Oracle Trace Discovery queries nodes for products capable of collecting Oracle Trace data.

Please select the nodes you would like queried. This may take a few minutes.

| Node | Status | Last Discovered |
|---|---|---|
| ✓ LGLOY... | | |
| ✓ trace | | |

The Oracle Trace Discovery dialog box contains the following fields and buttons:

**Node**
Select the nodes on which you want to perform the discovery. Only nodes that have Oracle Enterprise Manager agents running on them are displayed in the dialog box. For each selected node, the discovery process identifies whether Oracle Trace is installed on the node and what products are available for data collection. For products to be available for data collection, they must be instrumented with Oracle Trace calls.

**Status**
This field shows the status of the last discovery process that took place on the node or the status of the current discovery.

**Last Discovered**
This field lists when the last discovery took place for this node.

**Continue**
When you click the Continue button, Oracle Trace begins the discovery process. The Product Discovery Status box displays. This box provides a visual cue to the progress of the discovery and the results of the process.

**Cancel**
Click the Cancel button to stop the discovery process.

**Help**
Click the Help button to get help on the discovery process.

**Select All**
Click the Select All button to choose all the available nodes.

**Clear All**
Click the Clear All button to deselect the nodes.

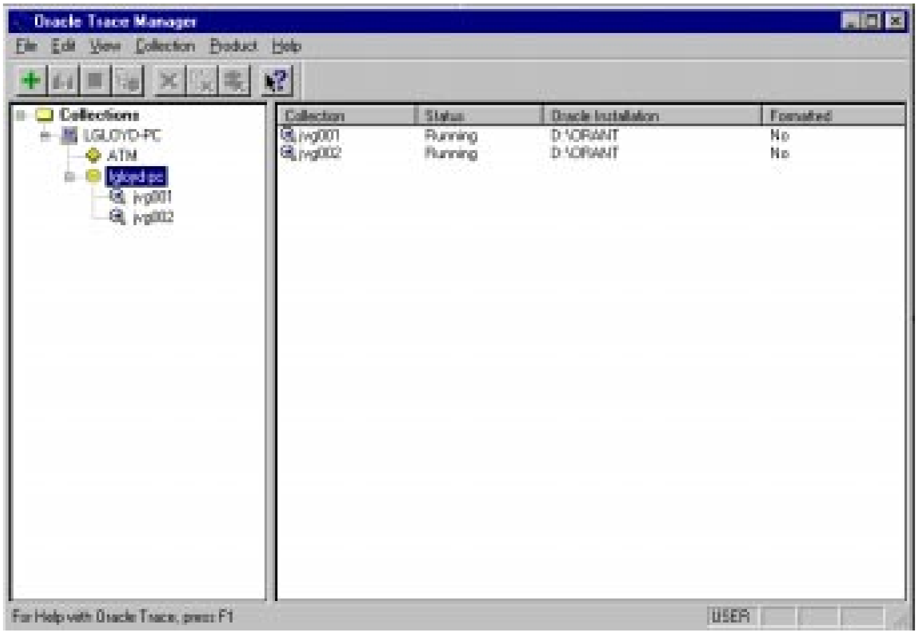
## Results of Discovery

After discovery has completed, you will see a list of discovered nodes and products. If you have created Oracle Trace collections on a node that fails Oracle Trace discovery, or if specific products or services fail discovery, the Oracle Trace navigator displays these collections and all associated products and services as disabled (grayed out).

# Viewing the Oracle Trace Manager Main Window

When you launch Oracle Trace, the Oracle Trace Manager main window displays, as shown in Figure 2–3.

*Figure 2–3 Oracle Trace Manager Main Window*



| Collection | Status | Oracle Installation | Formatted |
|---|---|---|---|
| jvg001 | Running | D:\ORANT | No |
| jvg002 | Running | D:\ORANT | No |

For Help with Oracle Trace, press F1

The window consists of menus, panes, a toolbar, and a status bar. Use the menus and the toolbar to access Oracle Trace functionality. The status bar at the bottom of the window provides information relating to the task you are performing.

## Left Pane

The left pane of the Oracle Trace main window provides a tree structure representing the hierarchy of the Oracle Trace objects. The top-most object in the tree structure is the Collections container. Expanding this container displays all of the nodes that have been discovered by Oracle Trace. Expanding each node displays the list of products and databases installed on that node that are available for Oracle Trace collection. Each product or database may contain one or more collections that have been executed, or are currently executing. Click on the plus sign (+) to expand the tree and the minus sign (-) to collapse the tree.

## Right Pane

The right pane of the window displays information relevant to the object highlighted in the left pane. In general, this information provides details about active and completed Oracle Trace collections for a selected object in the tree. The information is based on data stored in Oracle Trace tables within the Oracle Enterprise Manager repository.

> **Note:** The repository is a set of tables stored in an Oracle database. For a more detailed description of the repository, refer to the *Oracle Enterprise Manager Concepts Guide.*

Some or all of the following information is provided in the right pane of the Oracle Trace main window, depending on what is highlighted in the left pane:

**Collection**
The collection name.

**Status**
The current status of the collection: created, running, completed, formatting, scheduled, stopping, stopped, deleting files, deleted files, and unknown.

**Node**
The name of the node on which the collection was run.

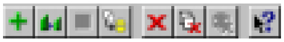
**Oracle Installation**
The location of the set of products from which data is being collected.

**Formatted**
The format status of the collection data. The status is listed as Yes if all of the data has been formatted. If the status is listed as No, then the data has not been formatted.

## Toolbar Icons

The Oracle Trace toolbar icons represent the items in the menus. The toolbar options (see Figure 2–4) are enabled depending on the objects selected in the tree list.

***Figure 2–4   Toolbar Icons***



The Oracle Trace icons are as follows:

Creates a collection. Click this icon to create a collection.

Creates a similar collection. Select a collection in the tree list, then click this icon to create a similar collection. This is the same as choosing the Collection=>Create Like... option.

Stops a collection. Select a collection in the tree list and click this icon to stop collecting event data.

Formats a collection. Select a collection in the tree list and click this icon to format a collection.

Deletes a collection from Oracle Trace Manager. Select a collection in the tree list and click this icon to delete the existing collection. This option also gives you the opportunity to delete the collection definition file, the data collection file, and formatted data.

Deletes the collection definition file (.cdf) and data collection file (.dat) for the selected collection.

Deletes formatted data. Select a collection with formatted data and click this icon to delete formatted data from the database.

Invokes Windows help mode. After clicking this icon, you can click the pulldown entry to see the object's context-sensitive help without invoking its command.

# Viewing Oracle Trace Manager Menus

The following sections briefly describe the menus in Oracle Trace Manager. The menus are:

- File
- Edit
- View
- Collection
- Product

- Help

## File Menu

The File menu contains the following option:

**Exit**
Exits the Oracle Trace application.

Collections that are currently running will continue to run until they are stopped, the maximum file size is reached (if one was specified when the collection was started), or the scheduled end time is reached.

## Edit Menu

The Edit menu contains the following options:

**Cut**
Removes highlighted text and places it in a buffer for the Paste operation. You can use this option in the Collection Wizard.

**Copy**
Makes a copy of highlighted text to be pasted in another location. You can use this option in the Collection Wizard.

**Paste**
Places copied or cut text at the position of the cursor. You can use this option in the Collection Wizard.

**Preferences...**
Displays the Preferences dialog box in which you specify default format options used by the Collection Wizard.

## View Menu

The View menu contains the following options:

**Refresh**
Forces Oracle Trace to update and validate against Oracle Enterprise Manager node and database information.

### Toolbar

Enables or disables display of the toolbar in the Oracle Trace Manager main window. A check next to the Toolbar option indicates that the toolbar is enabled.

### Status Bar

Enables or disables display of the Status bar in the Oracle Trace Manager main window. The Status bar displays status messages as actions are performed. A check next to the Status Bar option indicates that status messages are enabled.

## Collection Menu

The Collection menu contains the following options:

### Create...

Starts the Collection Wizard. Use this wizard to enter the information that Oracle Trace needs to create a collection.

### Create Like...

Starts the Collection Wizard if a collection is highlighted in the left pane of the Oracle Trace Manager main window. The information from the highlighted collection is copied to the new collection. You can either accept or change this information.

### Stop

Stops an active collection selected in the tree list.

### Format...

Formats the collection data to a database.

### View Formatted Data...

Starts the Oracle Trace Data Viewer application. This option is enabled when the collection highlighted in the right pane is a formatted collection.

### Delete...

Deletes a collection from Oracle Trace Manager. Select a collection in the tree list and choose this option to delete the existing collection. This menu item also gives you the option to delete the collection definition file, the data collection file, and formatted data.

### Delete Formatted Data...

Deletes formatted data. Select a collection with formatted data and click this option to delete formatted data from the database.

**Delete Collection Files...**
Deletes the collection definition file (.cdf) and the data collection file (.dat) for the selected collection.

## Product Menu

The Product menu contains the following options:

**Discovery...**
Starts a process that identifies whether Oracle Trace is installed on a node, and if so, identifies what, if any, products enabled with Oracle Trace API calls are on that node.

> **Note:** Oracle Trace Manager uses the list of nodes known to the Oracle Enterprise Manager console.

**Create**
Creates a new product definition file.

**Edit**
Edits an existing product definition file.

> **Note:** Your Oracle Trace Manager main window may not show the Product=>Create and the Product=>Edit menu options. These menu options contain functions needed only by application developers who will be creating new Oracle Trace product definition files. Therefore, the menu options are suppressed by default. For more information about invoking these menu options, see the *Oracle Enterprise Manager Oracle Trace Developer's Guide*.

## Help Menu

The Help menu contains the following options:

**Contents**
Provides a list of Oracle Trace help topics.

**Search for Help On...**
Displays a dialog box containing a scrollable list of index entries and keywords that you can search for in the help system.

### Using Help

Displays information on using the Microsoft Windows help system.

### About Oracle Trace Manager

Displays a dialog box containing version and copyright information about Oracle Trace Manager.

# 3

# Working with Collections

This chapter describes how to work with Oracle Trace collections. In particular, the following topics are covered:

- *Creating Collections*
- *Viewing Collection Definition and Properties*
- *Creating Like Collections*
- *Stopping Collections*
- *Formatting Collections*
- *Deleting Collection Entries*
- *Deleting Formatted Data*
- *Deleting Collection Files*
- *Running Reports*

## Creating Collections

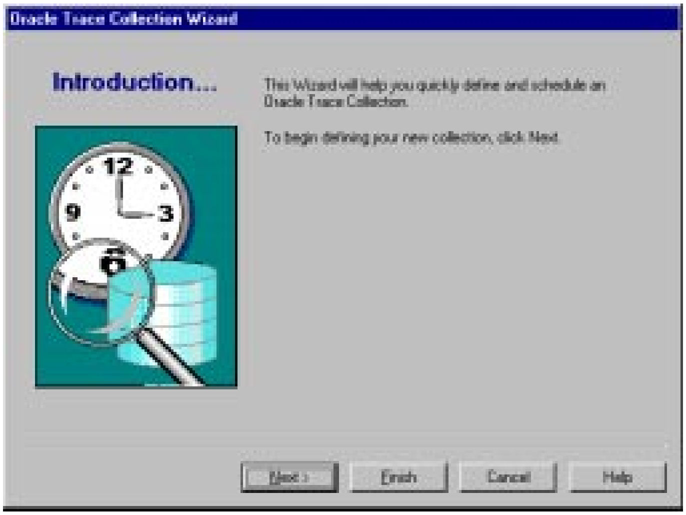
You create Oracle Trace collections using the Oracle Trace Collection Wizard. The Collection Wizard helps you define and schedule an Oracle Trace collection. To access the wizard, do one of the following:

- Press the plus sign (+) in the toolbar.
- Select Collection=>Create.
- Click the right mouse button after you select a node or product in the tree and select Create Collection.

The Collection Wizard prompts you only for the information it needs. This means that creating a collection is context-sensitive. If the node selection has already been established, you will skip to step 3.

If you are at the top of the tree list (the word Collection is highlighted) and you click the Create Collection toolbar button, the Introduction page of the Oracle Trace Collection Wizard displays, as shown in Figure 3–1.

***Figure 3–1 Introduction Page of Oracle Trace Collection Wizard***



This Wizard will help you quickly define and schedule an Oracle Trace Collection.

To begin defining your new collection, click Next.

The information that you enter in the Collection Wizard pages defines the collection parameters. Each step in the wizard gathers information based on the input from the previous step. The Collection Wizard prompts you for information in the following order:

1. Server node
2. Oracle Installation
3. Event sets
4. Restrictions on data collected
5. Output (collection definition files and data files)
6. Collection schedule

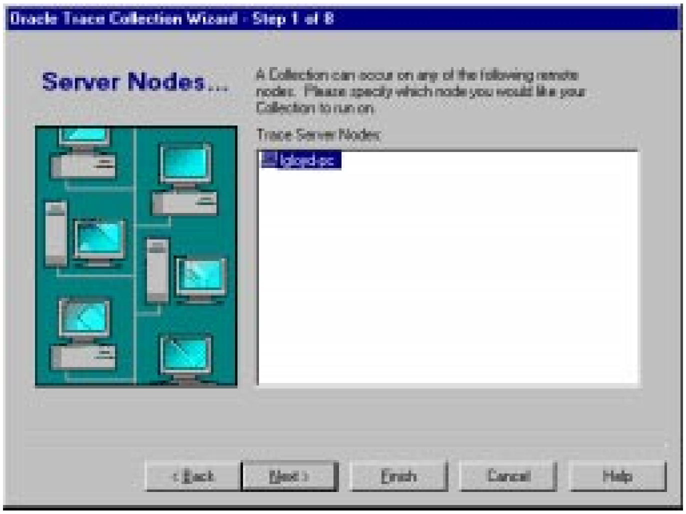
7. Advanced scheduling options
8. Formatting collected data
9. Advanced formatting options

Click the Next button to start the definition of the collection.

## Step 1: Select Server Nodes

In step 1, you select the server node (see Figure 3–2). If the server node can be determined by the context in which the Create Collection was activated, the Collection Wizard skips this page.

*Figure 3–2 Step 1 - Server Node*



Oracle Trace Collection Wizard - Step 1 of 8

Server Nodes...

A Collection can occur on any of the following remote nodes. Please specify which node you would like your Collection to run on.

Trace Server Nodes:

lploed-pc

**Trace Server Nodes**

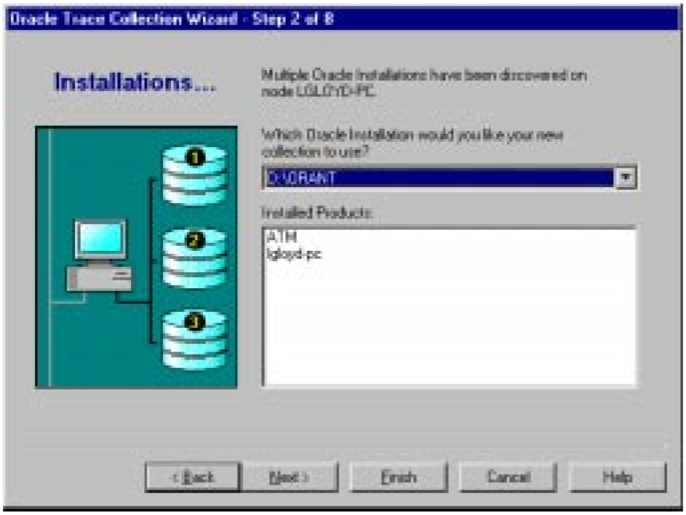
These are the server nodes that contain valid Oracle installations for running Oracle Trace collections. Select from this list to identify the node where the collection is to be run.

## Step 2: Choose Oracle Installation

In step 2, you choose which Oracle installation you would like the collection to use (see Figure 3–3).

You see this page only if Oracle Trace has been installed in more than one $ORACLE_HOME on the node. Otherwise, you will not be prompted with this page. The discovery mechanism locates all $ORACLE_HOME installations on a node. Each $ORACLE_HOME installation represents a different version of the Oracle products.

*Figure 3–3 Step 2 - Oracle Installation*



Oracle Trace Collection Wizard - Step 2 of 8

Installations...

Multiple Oracle Installations have been discovered on node LGLOYD-PC.

Which Oracle Installation would you like your new collection to use?

D:\ORANT

Installed Products:

ATM
lgloyd-pc

Also, an Oracle installation contains a compatible set of instrumented products. Oracle Trace collections can run on multiple products; however, the products must all belong to one Oracle installation.

**Installation**

This list shows all available Oracle installations for the selected node.

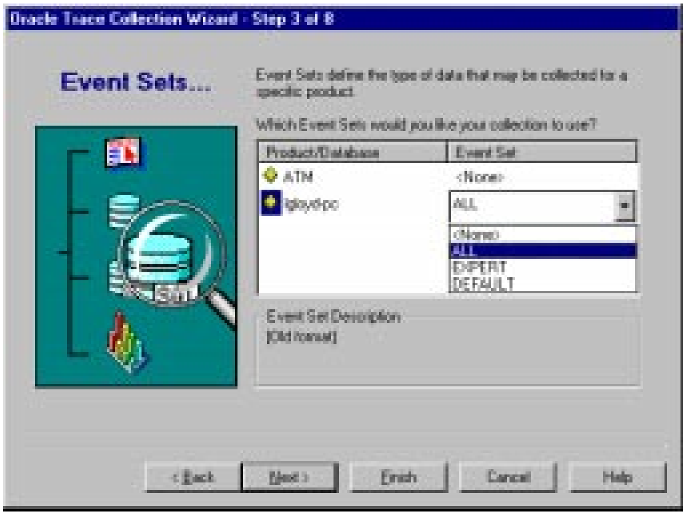
**Installed Products**

This is a read-only list showing the products available for the currently selected installation.  Only instrumented installed products are listed.

## Step 3: Select Event Sets

In step 3, you select the event sets that you want to use (see Figure 3–4). An event set is a grouping of events.

*Figure 3–4 Step 3 - Event Sets*



Oracle Trace Collection Wizard - Step 3 of 8

Event Sets...

Event Sets define the type of data that may be collected for a specific product

Which Event Sets would you like your collection to use?

| Product/Database | Event Set |
|---|---|
| ATM | <None> |
| lgloyd-pc | ALL |

Event Set Description
[Old format]

### Product/Database

This is a list of available products and databases that are instrumented for the selected node and Oracle installation. Event sets can be chosen from one or more products and databases to define the collection. Only one event set can be chosen per product or database.

### Event Set

This is a list of event sets for the particular product or database. You may select one event set per product or database, or no event sets if the product or database is not to be part of the collection. Select the event set value you want to use. For Oracle Server, Oracle provides the following event sets: ALL, DEFAULT, EXPERT (Oracle Expert application), CACHEIO (buffer cache I/O), and SUMMARY (Summary Advisor application).

> **Note:** The CACHEIO and SUMMARY event sets are only available with the Oracle Server release 8.1.5.

**Note:** Oracle Corporation recommends that you use either the EXPERT or DEFAULT event set when collecting Oracle Server data. The ALL event set generates large amounts of data as it collects wait events. Unless you specifically want to examine wait events, avoid using the ALL event set.

### Event Set Description

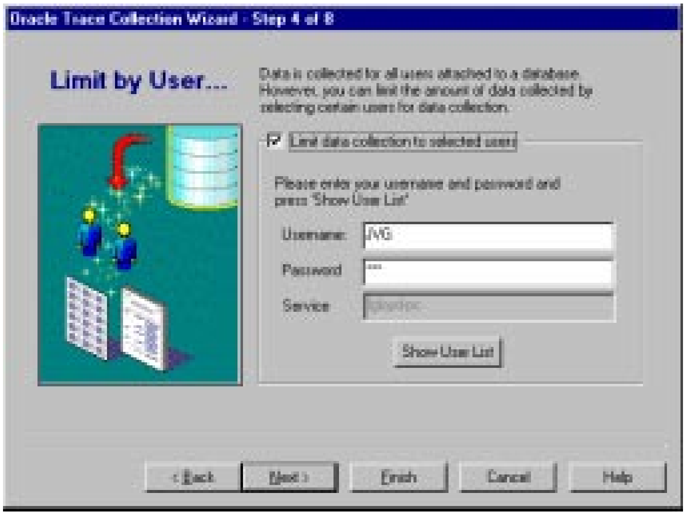
The Event Set Description describes the event set selected. [Old format] means that no description is supplied in the product definition file (.fdf).

## Step 4: Limit Data Collected by Users

By default, Oracle Trace collects data for all users attached to a database. In step 4, you can restrict the amount of data collected by selecting certain users for data collection (see Figure 3–5).

**Note:** This step is available only if you select a database on the Event Sets page and the database is on Oracle8 release 8.0.4 or higher. Otherwise you will not see this page.

*Figure 3–5 Step 4 - Limit by User*



Oracle Trace Collection Wizard - Step 4 of 8

Limit by User...

Data is collected for all users attached to a database. However, you can limit the amount of data collected by selecting certain users for data collection.

☑ Limit data collection to selected users

Please enter your username and password and press 'Show User List'

Username: JVG

Password: ***

Service: [illegible]

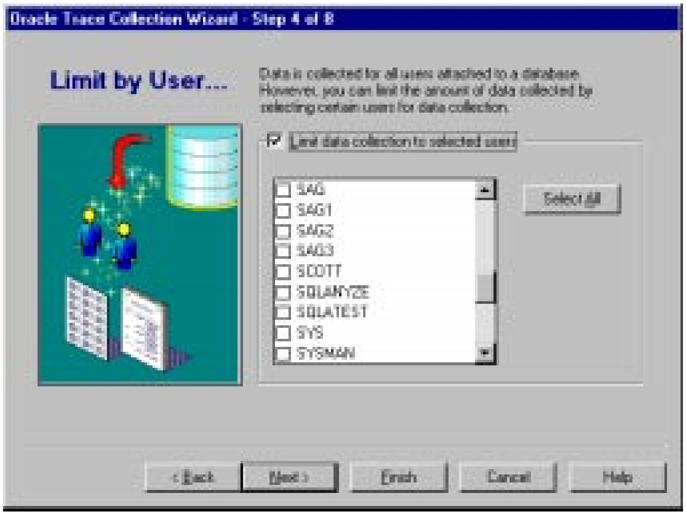
**Limit data collection to selected users**
Check this box if you want to limit the collection to certain users. Otherwise, all users will participate in the collection.

**Username and Password**
Supply this information to access the list of users on the target Oracle Server. This action allows Oracle Trace Manager to connect to the server to obtain the user names. This account must have the privileges required to select from the SYS.ALL_USERS table and the V$EVENT_NAME view.

**Service**
This field is grayed out. The service was established on the Event Sets page. If you need to change the service name, go back to the Event Sets page and select another service.

**Show User List**
Click this button to display a list of users, as shown in Figure 3–6.

*Figure 3–6 Step 4 - Limit by User (List)*



Oracle Trace Collection Wizard - Step 4 of 8

Limit by User...

Data is collected for all users attached to a database. However, you can limit the amount of data collected by selecting certain users for data collection.

☑ Limit data collection to selected users

☐ SAG
☐ SAG1
☐ SAG2
☐ SAG3
☐ SCOTT
☐ SQLANYZE
☐ SQLATEST
☐ SYS
☐ SYSMAN

Select the users that you want to participate in the data collection. Select a user by clicking in the square to the left of the user name. Selected users will have a checkmark in the square. Click the Select All button if you want all users to participate in the collection. This is the equivalent of not limiting the data collected. Once you click the Select All button, you can deselect all the choices by clicking the Deselect All button.

## Step 5: Limit Data Collected by Wait Events

By default, Oracle Trace collects data for all wait events. In step 5, you can restrict the amount of data collected by selecting certain wait events for data collection (see Figure 3–7).

---

**Note:** This step has a number of restrictions, as stated in the following list.

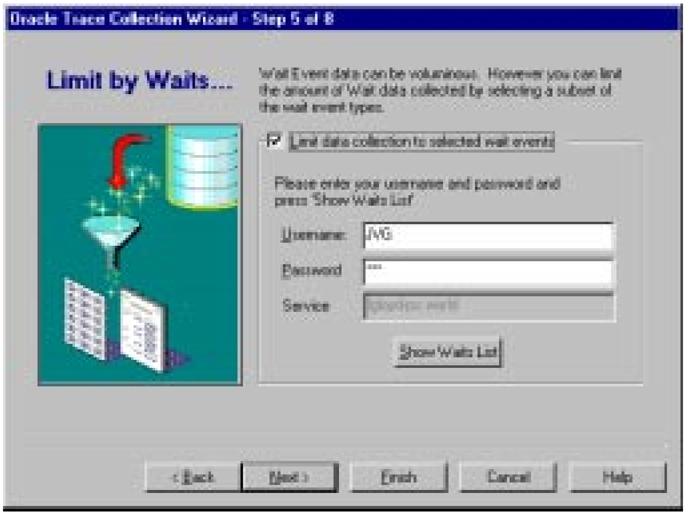
---

This step is only available if all the following conditions are true:

- You selected a database from the Product/Database column on the Event Sets page.
- The selected database is on Oracle8 release 8.0.4 or higher.

- The event set that you have chosen contains wait events, for example, the ALL event set.

*Figure 3–7 Step 5 - Limit by Waits*



Oracle Trace Collection Wizard - Step 5 of 8

Limit by Waits...

Wait Event data can be voluminous. However you can limit the amount of Wait data collected by selecting a subset of the wait event types.

☑ Limit data collection to selected wait events

Please enter your username and password and press 'Show Waits List'

Username: JVG

Password: ***

Service: [illegible]

### Limit data collection to selected wait events

Check this box if you want to limit the collection to certain wait events. Otherwise, all wait events will be collected.

### Username and Password

Supply this information to access the list of wait events on the server. This action allows you to connect to the server. This account must have the privileges required to select from the V$EVENT_NAME view.

> **Note:** If you supplied this information in step 4, you are not prompted for the information again.

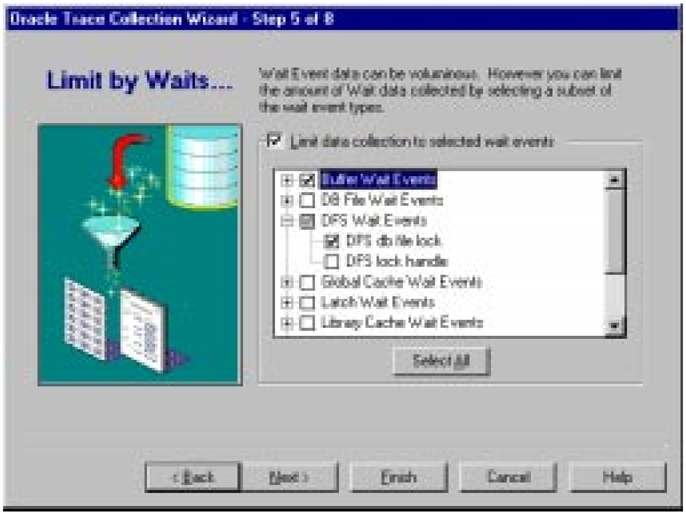
### Service

This field is grayed out. The service was established on the Event Sets page. If you need to change the service name, go back to the Event Sets page and select another service.

### Show Waits List

Click this button to display a list of wait events, as shown in Figure 3–8.

*Figure 3–8 Steps 5 - Limit by Waits (List)*



Oracle Trace Collection Wizard - Step 5 of 8

Limit by Waits...

Wait Event data can be voluminous. However you can limit the amount of Wait data collected by selecting a subset of the wait event types.

☑ Limit data collection to selected wait events

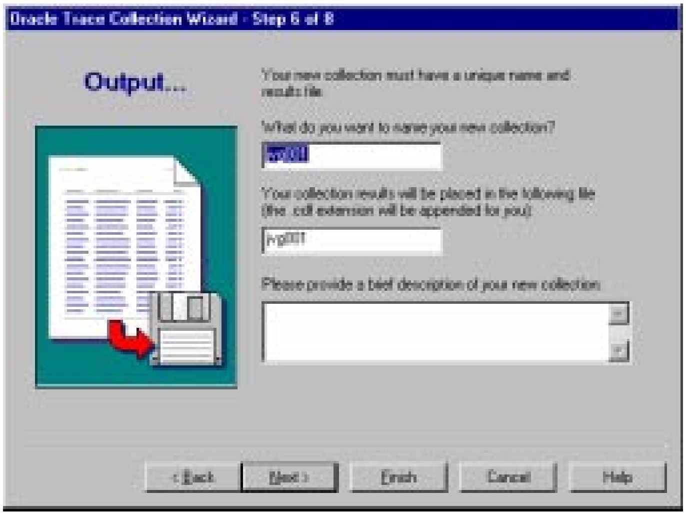
- ⊞ ☑ Buffer Wait Events
- ⊞ □ DB File Wait Events
- ⊟ ▣ DFS Wait Events
  - ☑ DFS db file lock
  - □ DFS lock handle
- ⊞ □ Global Cache Wait Events
- ⊞ □ Latch Wait Events
- ⊞ □ Library Cache Wait Events

Select the wait events that you want to participate in the data collection. Wait events are grouped by wait event types. Expand the tree to view individual wait events. You can click on particular wait events or wait event types. Select a wait event by clicking in the square to the left of the wait event. Selected wait events will have a checkmark in the square. Click the Select All button if you want all wait events to participate in the collection. This is the equivalent of not limiting the data collected. Once you click the Select All button, you can deselect all the choices by clicking the Deselect All button.

## Step 6: Name Collection Definition File

In step 6, Oracle Trace provides a default name of the collection definition file in which to store the results of the collection (see Figure 3–9). You can either accept the default or change the name.

*Figure 3–9 Step 6 - Output*



Oracle Trace Collection Wizard - Step 6 of 8

Output...

Your new collection must have a unique name and results file.

What do you want to name your new collection?

jvg001

Your collection results will be placed in the following file (the .cdl extension will be appended for you):

jvg001

Please provide a brief description of your new collection:

### Collection Name

If you do not accept the default name, specify a name of up to 16 characters for the collection. Special symbols are not allowed. Both the collection name and the collection definition file name must be unique on the node where you are collecting the data.

### Collection Results File

By default, the collection file has the same name as the collection with .cdf (collection definition file) appended. The complete file specification is $ORACLE_HOME/otrace/admin/cdf/*name*.cdf where *name* is what was in the Collection Name field; however, this name may be different on systems other than UNIX. The name can be up to 8 characters in length.

### Description

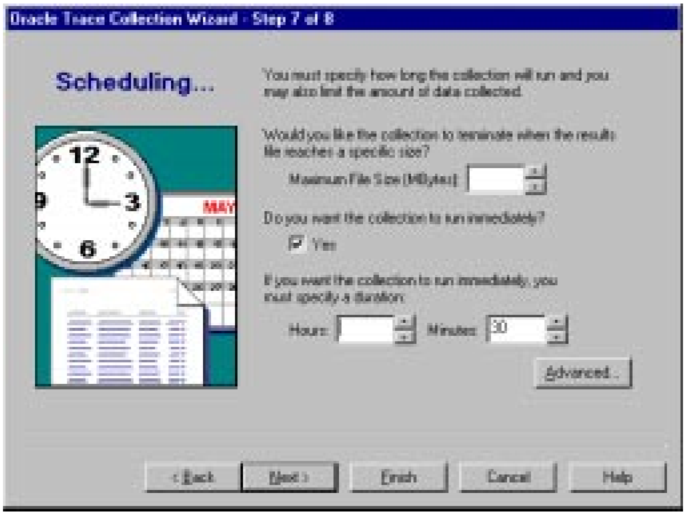
You also have the option of providing a description of this new collection.

At this point, you can click the Finish button. If you do, Oracle Trace accepts all of the defaults for scheduling and formatting information. These will be displayed on the Summary page (see Figure 3–14).

## Step 7: Schedule the Collection

In step 7, you have the opportunity both to define the maximum length (in megabytes) of the collection definition file (.cdf) and to provide scheduling information (see Figure 3–10).

*Figure 3–10 Step 7 - Scheduling*



Oracle Trace Collection Wizard - Step 7 of 8

Scheduling...

You must specify how long the collection will run and you may also limit the amount of data collected.

Would you like the collection to terminate when the results file reaches a specific size?

Maximum File Size (MBytes):

Do you want the collection to run immediately?

☑ Yes

If you want the collection to run immediately, you must specify a duration:

Hours: Minutes: 30

### Maximum File Size

If you specify a maximum file size, the collection will terminate once the file size is reached, even if other scheduling criteria have not been met. Oracle Corporation recommends that you set this parameter if you have a disk space limitation. Use a value from 1 through 2000 inclusive. The default is no maximum file size.

> **Note:** When you select the ALL event set for an Oracle Server collection, large amounts of data are collected, especially wait event data. In this case, Oracle Corporation highly recommends that you set a maximum file size.

### Collection to Run Immediately?

Checking this box signifies that you want the collection to run immediately. This is the default. Use the Advanced Scheduling Options page to schedule the collection if the collection is not to run immediately.

### Hours

Use a whole number to define this parameter.

**Note:** Collections running for hours consume a considerable amount of disk space.

### Minutes

Use a whole number to define this parameter. The default is 30 minutes.

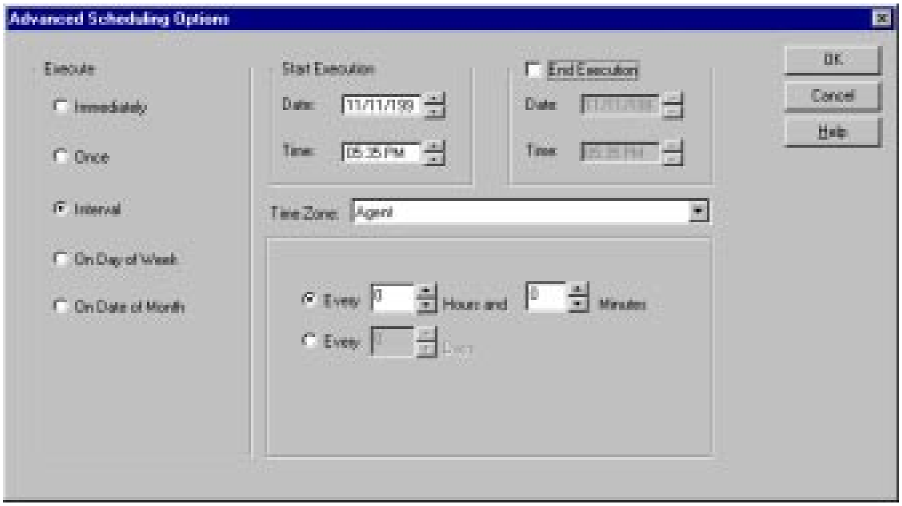
### Advanced...

Click the Advanced... button to display the advanced scheduling options, as described in the next section. If you do not choose to run the collection immediately, you need to supply the advanced scheduling options.

## Advanced Scheduling Options

The Advanced Scheduling Options page allows you to schedule the execution of the collection (see Figure 3–11).

*Figure 3–11 Advanced Scheduling Options Page*



Advanced Scheduling Options

Execute
○ Immediately
○ Once
● Interval
○ On Day of Week
○ On Date of Month

Start Execution
Date: 11/11/199
Time: 05:35 PM

□ End Execution
Date: 11/11/1998
Time: 05:35 PM

Time Zone: Agent

● Every 0 Hours and 0 Minutes
○ Every 0 Days

### Execute

Select the frequency with which you want the collection executed. The lower right pane changes format depending on the choice in the Execute field. The choices are:

- **Immediately**

  Schedules the collection as soon as you click the Submit button on the Summary page. The collection executes only one time.

- **Once**

  Schedules the collection only one time at the date and time you choose.

- **Interval**

  Allows you to schedule a specific time interval between collection executions. The interval can be a combination of hours and minutes, or number of days. Click on the scroll buttons to change the values or type in new values.

- **On Day of Week**

  Allows you to schedule the collection on one or more days (Sunday, Monday, etc.) of the week. Click on the days of the week in the Date field to select the days on which you want the collection to occur.

- **On Date of Month**

  Allows you to schedule the collection on one or multiple days (1-31) of the month. Click on the dates of the month in the Date field to select the dates on which you want the collection to occur.

### Start Execution

Choose the first date and time that you want the collection executed. This is the starting time for any collection scheduled on an interval.

- **Date**

  Select the month, day, or year in the Date field and click on the scroll buttons to change the value. You can also type in new values.

- **Time**

  Select the hour, minute, or AM/PM in the Time field and click on the scroll buttons to change the value. You can also type in new values.

### End Execution

Choose the last date and time that you want the collection executed. This option does not apply if you chose the Immediately or Once execution options.

- **Date**

  Select the month, day, or year in the Date field and click on the scroll buttons to change the value. You can also type in new values.

- **Time**

  Select the hour, minute, or AM/PM in the Time field and click on the scroll buttons to change the value. You can also type in new values.

### Time Zone

Select the time zone from the pull-down list. The choices are:

- **Agent**

  The agent schedules the collection execution based on the system time of the destination node.

- **Console**

  The agent schedules the collection execution simultaneously on all destinations based on the system time of the console.

- **GMT**

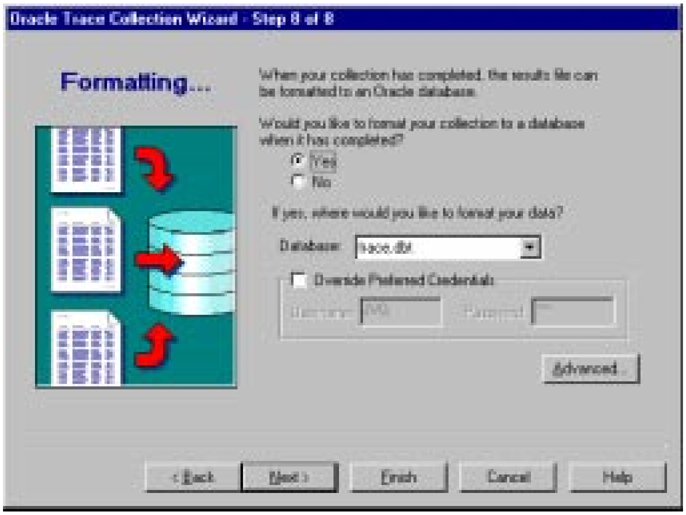
The agent schedules the collection execution simultaneously on all destinations based on Greenwich mean time (GMT).

## Step 8: Select Formatting Option

In step 8, you have the option of automatically formatting your collection data to an Oracle database when the collection ends (see Figure 3–12).

For Oracle Server releases prior to 7.3.4 and 8.0.4, Oracle Trace tables must be created before formatting; for details, see "Formatting Collections" on page 3-22.

*Figure 3–12 Step 8 - Formatting*



Oracle Trace Collection Wizard - Step 8 of 8

Formatting...

When your collection has completed, the results file can be formatted to an Oracle database.

Would you like to format your collection to a database when it has completed?

● Yes
○ No

If yes, where would you like to format your data?

Database: trace.dbt

□ Override Preferred Credentials

Username:
Password:

**Yes**
Choose Yes if you want to format your collection to a database when the collection is completed. This is the default.

**No**
Choose No if you do not want to format your collection to a database when the collection is completed.

**Database**
If you chose Yes, provide the name of the database where you want your formatted data to reside.

### Override Preferred Credentials

Preferred credentials identify the user name and password used to connect to the database and to load data into the Oracle Trace formatter tables. You can override the Oracle Enterprise Manager console or Oracle Trace preferred credentials by selecting the Override Preferred Credentials option. Select a database and enter the criteria for the database user name that is the target for the formatter tables.

Oracle Trace determines the preferred credentials in the following order. The search continues until one of the criteria is met.

1. Oracle Trace preferences by using Oracle Trace Edit=>Preferences
2. Oracle Enterprise Manager preferred credentials for that node

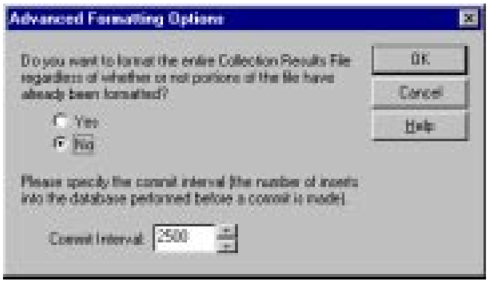
### Advanced...

Click the Advanced button to display the advanced formatting options, as described in the next section.

## Advanced Formatting Options

In the Advanced Formatting Options dialog box, you have the opportunity to have Oracle Trace either format the entire collection results file or format only unformatted data. See Figure 3–13.

*Figure 3–13   Advanced Formatting Options Page*



Advanced Formatting Options

Do you want to format the entire Collection Results File regardless of whether or not portions of the file have already been formatted?

○ Yes
● No

Please specify the commit interval (the number of inserts into the database performed before a commit is made).

Commit Interval: 2500

### Yes

Choose this option if you want to format the entire collection results file.

If you have already formatted part of the data and then you format the entire results, some event data will be formatted twice. The partial format and the full format will have different collection IDs. Make sure you query only the full format for consistent results. If you do not ask for the entire file, you get only the records that were not already formatted.

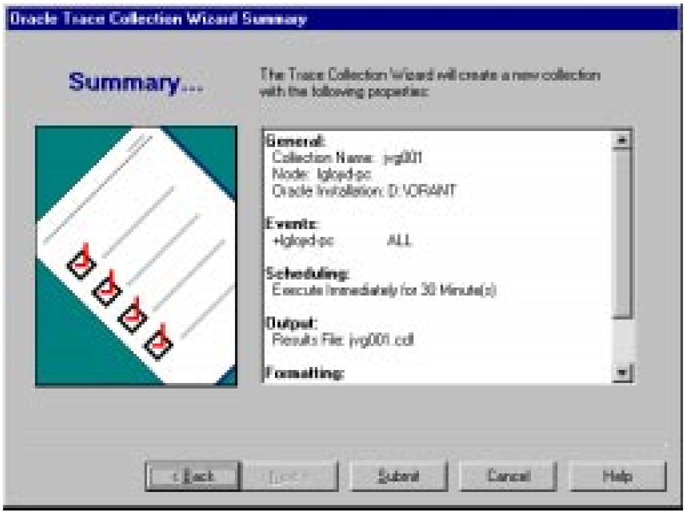
**No**
Choose this option if you want to format only unformatted data. This is the default. By specifying this option, you can examine data as it is collected. You do not have to wait for the entire collection to finish before you can look at the data. Use this option if the collection is scheduled to execute more than once.

**Commit Interval**
You can define the commit interval into the database. The shorter the interval, the longer it takes the collection to format. Also, the larger the commit interval, the larger the rollback segment needs to be for the commit operation to succeed. The default is 2500 inserts before a commit is made.

## Summary Page

The Oracle Trace Collection Wizard Summary page summarizes all of the information that you entered while using the Oracle Trace Collection Wizard (see Figure 3–14).

***Figure 3–14 Summary Page***



The Trace Collection Wizard will create a new collection with the following properties:

General:
Collection Name: jvg001
Node: lgloyd-pc
Oracle Installation: D:\ORANT

Events:
+lgloyd-pc ALL

Scheduling:
Execute Immediately for 30 Minute(s)

Output:
Results File: jvg001.cdf

Formatting:

The summarized properties are:

- General
- Events
- Limit Data Collected (only appears if restrictions were made)
- Scheduling
- Output
- Formatting

If, while reviewing the Summary page, you find options that you want to change, click the Back button to make the necessary changes.

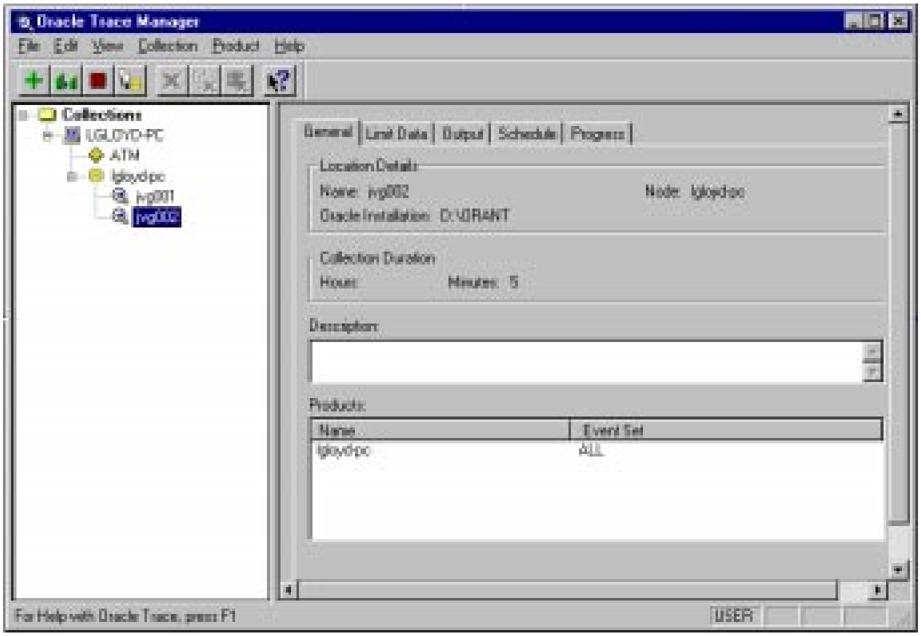
Once you are satisfied with the information, click the Submit button to start the collection process.

## Viewing Collection Definition and Properties

Once you have created a collection, you can view the collection properties, but you cannot edit the collection. However, if you need to create a similar collection, use the Collection=>Create Like... option or click the Create Like... button. Change the collection options needing to be updated, and start the new collection.

To view the collection properties, click the name of the collection in the tree. Oracle Trace displays the property pages of the collection: General, Limit Data, Output, Schedule, and Progress (see Figure 3–15).

*Figure 3–15 Collection Property Pages*



Location Details

Name: jvg002

Node: lgloydpc

Oracle Installation: D:\ORANT

Collection Duration

Hours:

Minutes: 5

Description:

Products:

| Name | Event Set |
|---|---|
| lgloydpc | ALL |

For Help with Oracle Trace, press F1

### General Page

The General page provides a minimal amount of information about collections. The information includes the location of the collection file, the duration of the collection, a description of the collection file, and the product and event set pairs that make up the collection. The fields on the General page are:

- Location Details
  - Name: Name of the collection
  - Oracle Installation: Location of the Oracle installation containing the product event sets
  - Node: Node on which the collection ran
- Collection Duration

  Length of time that the collection will run (or ran)
- Description

  Collection description that you provided

- Products

  These fields describe data that was collected:

  - Name: Name of the product or service for which data was collected
  - Event Set: The particular event set within the product for which data was collected

### Limit Data Page

The Limit Data page lists the items and events used as restrictions for data collection. If no restrictions were chosen, the text "No limits specified" displays.

### Output Page

The Output page displays the name of the .cdf file and the format history. The fields on the Output page are:

- Result File Details
  - Name: Name of the collection definition (.cdf) file where collected data is located
  - Maximum Size (megabytes): Size specified when the collection was defined

  **Note:** If no maximum size is present, the size is unlimited.

- Format History

  A record is created every time data is formatted.  Displays the database destination of the formatted data.

  - Database: Service name
  - User: Username supplied when data was formatted
  - Status: Shows the success/failure of the format
  - Notification Time: Shows the date and time that the data was formatted.
  - Partial: Yes indicates part of the data was formatted, that is, only new data collected since the last format, if any. If the field contains No, all of the data was formatted.

### Schedule Page

The Schedule page shows the type of execution for the collection, and in some cases, the start date/time of the collection, the end date/time of the collection, the time zone used for the collection, and the duration of the collection. These were defined if you chose the advanced scheduling options when you created the collection. Refer to "Advanced Scheduling Options" on page 3-13 for a detailed explanation of all these options.

### Progress Page

The Progress page displays the status of the collection as it was running, including the notification time. The fields on the Progress Collection Status page are:

- Action: Oracle Trace behavior; for example, creating a collection, formatting a collection, and so on.
- Status: State of the action; for example, starting collection, stopping collection, and so on.
- Notification Time: The time that Oracle Trace was notified of a change in status.

## Creating Like Collections

This option allows you to use existing collections as templates for new collections. This makes it easier for you to schedule collections by using a previous collection and changing only a few items, rather than specifying everything, every time.

In the collection tree list of the Oracle Trace main window, click the collection on which you want to base your new collection. Choose the Collection=>Create Like... option. This starts the Collection Wizard. Make the changes you need, following the steps previously outlined in this chapter.

## Stopping Collections

Stopping a collection terminates the data collection. To stop a collection:

1. Select a collection name from the Oracle Trace Manager main window.
2. Choose Collection=>Stop or select the Stop Collection icon in the toolbar.

Stopped collections have a status of “Stopped” in the Oracle Trace Manager main window.

## Formatting Collections

Oracle Trace collects the event data in a binary file for efficiency.  This binary file can be automatically formatted to an Oracle database for SQL access and reporting. You can format that data by using Oracle Trace Manager.

After formatting the collected data, you can perform queries against your Oracle Trace data using Data Viewer, or use it with a reporting tool such as Oracle Reports. You can also import Orace Trace formatted data into Oracle Expert to be used as SQL workload.

## Preparing to Format Oracle Trace Data

Before you format data to an Oracle database, formatter tables must be created in that database. Create formatter tables using the vobsh command, as described in the *Oracle Enterprise Manager Configuration Guide*.

> **Note:** The previous step is required for releases prior to Oracle Trace Collection Services release 7.3.4 and 8.0.4.

It is possible to maintain multiple sets of formatted data by creating formatter tables under more than one Oracle schema.

You can also format multiple collections into the same schema. You can format Oracle Trace data using either Oracle Trace Manager or the command-line interface. "Formatting Oracle Trace Data to Oracle Tables" on page A-7 provides information about using the command-line interface to format tables. The *Oracle Enterprise Manager Oracle Trace Developer's Guide* gives a complete description of the formatting steps using the command-line interface.

## Formatting a Collection Using Oracle Trace Manager

To format a collection using Oracle Trace Manager, select a collection name from the Oracle Trace Collection Summary window. Choose Collection=>Format from the Oracle Trace Manager main window. The Format Collection dialog box appears. Enter information as follows:

### Database

Enter a connect string or service name for the database.  If you enter nothing, Oracle Trace will use the default database.

### Override Preferred Credentials

You can override preferred credentials. These credentials identify the schema where the data will be loaded.

Oracle Trace determines the preferred credentials in the following order. The search continues until one of the criteria is met.

1. Oracle Trace preferences.
2. Oracle Enterprise Manager preferred credentials for that node.

**Username**
Enter the Oracle username under which the Oracle Trace formatter tables were created.

**Password**
Enter the password for the Username.

**Format Options**
Decide the commit interval that you want and whether you want a partial format of the data; that is, to process data that has not previously been formatted.

**Commit Interval**
Enter a number. The Commit Interval is the number of inserts made into the database before a commit is performed. A value is required in this field. The default is 2500.

**Partial Format**
Choose this option only if you want to process data that has not previously been formatted. This is useful when you want to format data for a collection while that collection is still in progress.

Click OK. When the collection is formatted, the status is listed in the Format column in the Collection Summary window.

## Deleting Collection Entries

To delete a collection entry from the Oracle Trace Manager main window, select the entry and choose Collection=>Delete. When you take this action, the Delete Collection dialog box appears.

The default is to delete everything, all collection files and all formatted data. However, by clicking the Advanced button, you can choose which formatted data you want to delete.

If you do not delete the collection files and the formatted data, Oracle Trace Manager deletes only its record of the collection. You will not see the General, Limit Data, Output, Schedule, and Progress pages for this collection. This also leaves the collection files and formatted data for you to delete manually at a later time.

For information about deleting formatted data for a collection, see the *Oracle Enterprise Manager Oracle Trace Developer's Guide*.

## Deleting Formatted Data

Because formatted data takes up a lot of space in the database, you may want to regularly delete formatted data. Deletion of formatted data is performed using the Oracle Trace Manager as follows:

1. Select a collection name from the Oracle Trace Collection Summary window.
2. Select the Collection=>Delete Formatted Data or the Delete Formatted Data icon in the toolbar. This displays a list of locations where data has been formatted to.

> **Note:** If data has been formatted a number of times to the same database, selecting one location will delete all formatted data relating to that collection in that selected database.

The Delete Formatted Data function deletes rows of data for a specific collection from the formatted database tables. Although it deletes the contents of event tables, it leaves the event tables themselves intact. This is because re-creating them would be time-consuming for the formatter.

> **Note:** The Delete Formatted Data function will not work correctly on formatter tables created with the version of otrcfmtc.sql that shipped with Oracle Server release 7.3.2 or earlier. If data already exists in old formatter tables and you wish to delete it, use the vobsh command. For more information, see the instructions for creating Oracle Trace formatter tables in the *Oracle Enterprise Manager Configuration Guide.*

## Deleting Collection Files

To delete the files associated with a collection, first select a collection name in the Oracle Trace Manager main window. Then choose Collection=>Delete Collection Files.

This deletes the collection definition file (.cdf) and the data collection (.dat) file. It does not delete the collection entry from the Oracle Trace Manager main window;

Deleted Files displays in the Status column. You cannot use Oracle Trace Manager to delete formatted data once the collection files have been deleted. You need to manually delete the formatted data from the database using Oracle Server Manager or Oracle SQL*Plus Worksheet.

```
delete from epc_collection
       where collection_id=<id>;
```

Deleting the collection’s record from epc_collection will also delete other related data records in other format tables.

## Running Reports

Another way to examine the collection data, rather than formatting it, is to use the Oracle Trace report utility. The report utility reads the event data directly from the data collection (.dat) file and creates a text listing of the events.  This approach is not recommended for large collections.  For large collections you should format the data.

Refer to the *Oracle Enterprise Manager Oracle Trace Developer’s Guide* for detailed information about using the report utility.

# 4

# Oracle Trace Data Viewer

Oracle Trace Data Viewer (hereafter referred to as Data Viewer) is an Oracle Enterprise Manager application that allows you to view formatted data collected by Oracle Trace.

This chapter includes the following:

- *Overview*
- *Usage Scenarios for Data Viewer*
- *Starting Data Viewer*
- *Using Data Viewer*
- *Using the Data View Wizard*

## Overview

An Oracle Trace server collection typically contains a large volume of valuable information that can be used for troubleshooting and investigating SQL or wait activity and resource utilization. Data Viewer handles the complex task of extracting data and aggregating key server performance metrics on a large Oracle Trace collection. Once you select an Oracle Trace collection, you can have Data Viewer compute SQL or Wait statistics or both.

Once you select an Oracle Trace collection, Data Viewer runs through all of the Oracle Trace formatter tables, extracting, processing, and aggregating key performance metrics. This processed data is presented in a comprehensive set of Oracle Trace predefined data views.

A **data view** is the definition of a query into the formatted data collected by Oracle Trace. A data view consists of items or statistics to be returned and, optionally, a sort order and limit of rows to be returned.

With the data views provided by Data Viewer, you can:

- Examine important statistical data, for example, elapsed times
- Drill down as needed to get additional details about the data

If you want to define your own data views, you can use the Oracle Trace Data View Wizard.

## Usage Scenarios for Data Viewer

Using Data Viewer you can:

- Look at Oracle Server SQL statistics

  Data Viewer provides predefined data views that can accelerate the process of identifying poorly performing SQL statements. Oracle Server SQL data views are grouped by categories. Some specific data views include:

  – Elapsed Time statistics

    Average Elapsed Time (sorts SQL by greatest average elapsed time per execution of queries within the collection)

  – Sort statistics

    Sorts in memory and sorts on disk and number of rows sorted

  – I/O statistics

    Disk Reads/Logical Reads Ratio (sorts SQL by worst disk to logical I/O data buffer cache hit rate)

- View Logical and Physical Transaction level statistics
- View Wait statistics

## Starting Data Viewer

You can access Data Viewer in the following ways:

- From within Oracle Trace
- From the Oracle Enterprise Manager program group
- From the Start menu

---

**Note:** Oracle Trace data must be formatted before you can use Data Viewer. Refer to Chapter 3 to learn how to format Oracle Trace data.

---

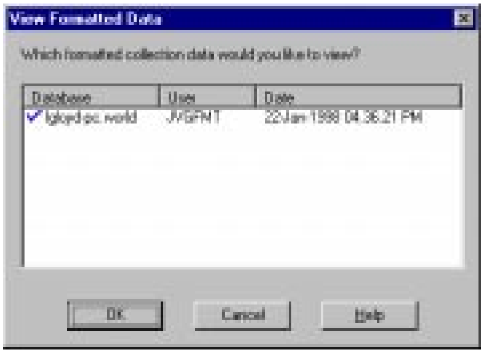
The following sections explain how to use these methods.

## Starting Data Viewer from Within Oracle Trace

To start Data Viewer from within Oracle Trace, do the following:

- Highlight a *formatted* collection in the navigator tree.
- Choose Collection=>View Formatted Data.
- Check the collection in the View Formatted Data dialog box (see Figure 4–1).
- Click OK.

***Figure 4–1 View Formatted Data Dialog Box***



View Formatted Data

Which formatted collection data would you like to view?

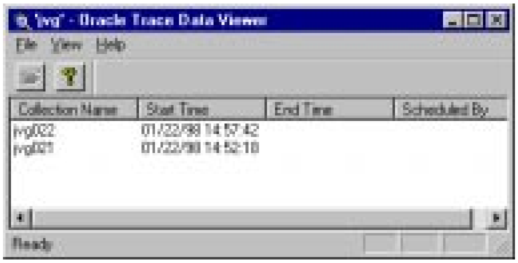
| Database | User | Date |
|---|---|---|
| ✓ lgloyd.oc.world | JVGPNT | 22-Jan-1998 04:36:21 PM |

After you click OK, the Data Viewer Work in Progress dialog box is displayed. The Work in Progress dialog box displays messages as Data Viewer checks database system clock granularity, creates indexes, and so on.

## Starting Data Viewer from Oracle Enterprise Manager

To start Data Viewer from Oracle Enterprise Manager, either click on Trace Data Viewer from the Diagnostics Pack drawer or select Tools=>Diagnostics Pack=>Trace Data Viewer. The Oracle Trace Data Viewer Login box appears. Provide login information to connect to a schema containing formatted collections. Data Viewer displays all collections formatted to the schema (see Figure 4–2).

***Figure 4–2 Data Viewer Collection Screen***



| Collection Name | Start Time | End Time | Scheduled By |
|---|---|---|---|
| jvg022 | 01/22/98 14:57:42 | | |
| jvg021 | 01/22/98 14:52:10 | | |

Decide which formatted collection you want to investigate and double-click on the collection name. The Work in Progress dialog box appears.

## Starting Data Viewer from the Start Menu

To start Data Viewer from the Start menu, select Start=>Programs=>ORACLE_HOME=>Diagnostics Pack=>Trace Data Viewer.

> **Note:** ORACLE_HOME represents the oracle_home in which the Diagnostics Pack is installed.

The Oracle Trace Data Viewer Login box appears. Provide login information to connect to a schema containing formatted collections. Data Viewer displays all collections formatted to the schema (see Figure 4–2).

Decide which formatted collection you want to investigate and double-click on the collection name. The Work in Progress dialog box appears.

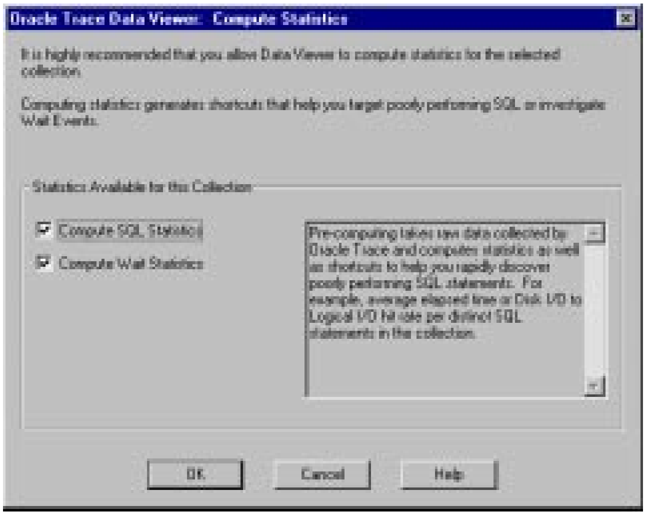
# Using Data Viewer

The first time a collection is accessed by Data Viewer, the Compute Statistics dialog box appears. At this time, Data Viewer can compute SQL and Wait statistics.

## Computing Statistics – Preparing Data for Fast Access and Analysis

The first time you run Data Viewer on a formatted collection containing Oracle Server data, Data Viewer asks whether you want to calculate SQL statistics, Wait statistics, or both (see Figure 4–3).

*Figure 4–3 Compute Statistics*



Oracle Trace Data Viewer: Compute Statistics

It is highly recommended that you allow Data Viewer to compute statistics for the selected collection.

Computing statistics generates shortcuts that help you target poorly performing SQL or investigate Wait Events.

Statistics Available for this Collection

☑ Compute SQL Statistics

☑ Compute Wait Statistics

Pre-computing takes raw data collected by Oracle Trace and computes statistics as well as shortcuts to help you rapidly discover poorly performing SQL statements. For example, average elapsed time or Disk I/O to Logical I/O hit rate per distinct SQL statements in the collection.

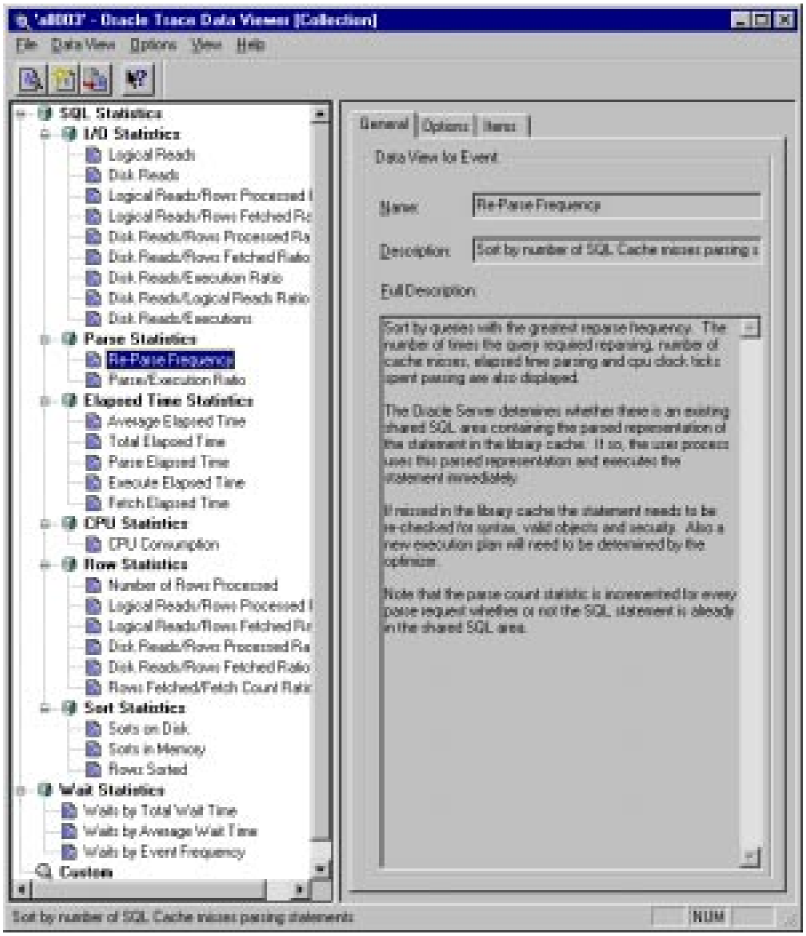
This step prepares the data for fast access and simplified problem analysis. Oracle Corporation ***strongly*** recommends that you allow this processing to occur.

**Note:** For large collections, the Computing Statistics step is time consuming. Oracle Corporation recommends that you perform this step at off-peak times.

## Displaying Data Views

Data Viewer displays the predefined data views for a collection, as shown in Figure 4–4.

*Figure 4–4 List of Predefined Data Views*



- SQL Statistics
  - I/O Statistics
    - Logical Reads
    - Disk Reads
    - Logical Reads/Rows Processed I
    - Logical Reads/Rows Fetched Ra
    - Disk Reads/Rows Processed Ra
    - Disk Reads/Rows Fetched Ratio
    - Disk Reads/Execution Ratio
    - Disk Reads/Logical Reads Ratio
    - Disk Reads/Executions
  - Parse Statistics
    - Re-Parse Frequency
    - Parse/Execution Ratio
  - Elapsed Time Statistics
    - Average Elapsed Time
    - Total Elapsed Time
    - Parse Elapsed Time
    - Execute Elapsed Time
    - Fetch Elapsed Time
  - CPU Statistics
    - CPU Consumption
  - Row Statistics
    - Number of Rows Processed
    - Logical Reads/Rows Processed I
    - Logical Reads/Rows Fetched Ra
    - Disk Reads/Rows Processed Ra
    - Disk Reads/Rows Fetched Ratio
    - Rows Fetched/Fetch Count Ratio
  - Sort Statistics
    - Sorts on Disk
    - Sorts in Memory
    - Rows Sorted
- Wait Statistics
  - Waits by Total Wait Time
  - Waits by Average Wait Time
  - Waits by Event Frequency
- Custom

Data View for Event

Name: Re-Parse Frequency

Description: Sort by number of SQL Cache misses parsing s

Full Description:

Sort by queries with the greatest reparse frequency. The number of times the query required reparsing, number of cache misses, elapsed time parsing and cpu clock ticks spent parsing are also displayed.

The Oracle Server determines whether there is an existing shared SQL area containing the parsed representation of the statement in the library cache. If so, the user process uses this parsed representation and executes the statement immediately.

If missed in the library cache the statement needs to be re-checked for syntax, valid objects and security. Also a new execution plan will need to be determined by the optimizer.

Note that the parse count statistic is incremented for every parse request whether or not the SQL statement is already in the shared SQL area.

Sort by number of SQL Cache misses parsing statements

Depending on the events collected, the predefined data view list will vary. For example, the Oracle Server collection will contain predefined data views appropriate for data collected in an Oracle Server collection.

---

**Note:** Predefined data views reflect statistics computed for all occurrences of a distinct query within the collection (see Table 4–1.)

---

If the Oracle Trace ALL class was used for the collection, Wait events will also display in the navigator tree. Unless you need to examine wait events, Oracle Corporation suggests that you collect data for the DEFAULT or EXPERT class to minimize the amount of data collected.

*Table 4–1 Predefined Data Views Provided By Oracle Trace*

| View Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Logical Reads** | Total number of logical reads performed for each distinct query. | Total number of blocks read during parses, executions and fetches.<br>Logical reads for parses, executions and fetches of the query. | Logical data block reads include data block reads from both memory and disk.<br>Input/output is one of the most expensive operations in a database system. I/O intensive statements can monopolize memory and disk usage causing other database applications to compete for these resources. |
| **Disk Reads** | Queries that incur the greatest number of disk reads. | Disk reads for parses, executions, and fetches. | Disk reads also known as physical I/O are database blocks read from disk. The disk read statistic is incremented once per block read regardless of whether the read request was for a multiblock read or a single block read. Most physical reads load data, index, and rollback blocks from the disk into the buffer cache.<br>A physical read count can indicate a high miss rate within the data buffer cache. |
| **Logical Reads/Rows Fetched Ratio** | Number of logical reads divided by the number of rows fetched for all executions of the current query. | Total logical I/O.<br>Total number of rows fetched. | The more blocks accessed relative to the number of rows actually returned the more expensive each row is to return.<br>Can be a rough indication of relative expense of a query. |
| **Disk Reads/ Rows Fetched Ratio** | Number of disk reads divided by the number of rows fetched for all executions of the current query. | Total disk I/O.<br>Total number of rows fetched. | The greater the number of blocks read from disk for each row returned the more expensive each row is to return.<br>Can be a rough indication of relative expense of a query. |
| **Disk Reads/ Execution Ratio** | Total number of disk reads per distinct query divided by the number of executions of that query. | Total disk I/O.<br>Logical I/O for the query as well as the number of executions of the query. | Indicates which statements incur the greatest number of disk reads per execution. |

*Table 4–1 Predefined Data Views Provided By Oracle Trace (Cont.)*

| View Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Disk Reads/ Logical Reads Ratio** | Greatest miss rate ratio of disk to logical reads. | Individual logical reads.<br>Disk reads for the query as well as the miss rate. | The miss rate indicates the percentage of times the Oracle Server needed to retrieve a database block on disk as opposed to locating it in the data buffer cache in memory.<br>The miss rate for the data block buffer cache is derived by dividing the physical reads by the number of accesses made to the block buffer to retrieve data in consistent mode plus the number of blocks accessed through single block gets.<br>Memory access is much faster than disk access; the greater the hit ratio, the better the performance. |
| **Re-Parse Frequency** | Queries with the greatest reparse frequency. | Number of cache misses.<br>Total number of parses.<br>Total elapsed time parsing.<br>Total CPU clock ticks spent parsing. | The Oracle Server determines whether there is an existing shared SQL area containing the parsed representation of the statement in the library cache. If so, the user process uses this parsed representation and executes the statement immediately.<br>If missed in the library cache, the statement needs to be re-checked for syntax, valid objects, and security. Also a new execution plan will need to be determined by the optimizer.<br>Note that the parse count statistic is incremented for every parse request whether or not the SQL statement is already in the shared SQL area. |
| **Parse/Execution Ratio** | Number of parses divided by the number executions per statement. | Individual number of parses.<br>Number of executions. | The count of parses to executions should be as close to one as possible. If there are a high number of parses per execution then the statement has been needlessly reparsed. This could indicate the lack of use of bind variables in SQL statements or poor cursor reuse.<br>Reparsing a query means that the SQL statement has to be re-checked for syntax, valid objects and security. Also a new execution plan will need to be determined by the optimizer. |
| **Average Elapsed Time** | Greatest average time spent parsing, executing and fetching on behalf of the query. | Individual averages for parse, execution and fetch. | The average elapsed time for all parses, executions and fetches-per-execution are computed, then summed for each distinct SQL statement in the collection. |
| **Total Elapsed Time** | Greatest total elapsed time spent parsing, executing and fetching on behalf of the query. | Individual elapsed times for parses, executions and fetches. | The total elapsed time for all parses, executions and fetches are computed, then summed for each distinct SQL statement in the collection. |

*Table 4–1 Predefined Data Views Provided By Oracle Trace (Cont.)*

| View Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Parse Elapsed Time** | Total elapsed time for all parses associated with a distinct SQL statement. | SQL cache misses.<br>Elapsed times for execution and fetching.<br>Total elapsed time. | During parsing the Oracle Server determines whether there is an existing shared SQL area containing the parsed representation of the statement in the library cache. If so, the user process uses this parsed representation and executes the statement immediately.<br>If missed in the library cache, the statement needs to be rechecked for syntax, valid objects and security. Also a new execution plan will need to be determined by the optimizer. |
| **Execute Elapsed Time** | Greatest total elapsed time for all executions associated with a distinct SQL statement. | Total elapsed time.<br>Individual elapsed times for parsing and fetching. | The total elapsed time of all execute events for all occurrences of the query within an Oracle Trace collection. |
| **Fetch Elapsed Time** | Greatest total elapsed time for all fetches associated with a distinct SQL statement. | Number of rows fetched.<br>Number of fetches.<br>Number of executions.<br>Total elapsed time.<br>Individual elapsed times for parsing and executing. | The total elapsed time spent fetching data on behalf of all occurrences of the current query within the Oracle Trace collection. |
| **CPU Statistics** | Total CPU clock ticks spent parsing, executing and fetching on behalf of the SQL statement. | CPU clock ticks for parses, executions and fetches.<br>Number of SQL cache misses and sorts in memory. | When SQL statements and other types of calls are made to an Oracle Server, a certain amount of CPU time is necessary to process the call. Average calls require a small amount of CPU time. However, a SQL statement involving a large amount of data, a runaway query, in memory sorts or excessive reparsing can potentially consume a large amount of CPU time.<br>Note that CPU time displayed is in terms of the number of CPU clock ticks on the operating system housing the database. |
| **Number of Rows Returned** | Greatest total number of rows returned during fetch for the SQL statement. | Number of rows returned during the fetch operation as well as the execution rows. | Targets queries that manipulate the greatest number of rows during fetching. May mean that high gains can be made by tuning row intensive queries. |
| **Rows Fetched/ Fetch Count Ratio** | Number of rows fetched divided by the number of fetches. | Individual number of rows fetched.<br>Number of fetches. | This ratio shows how many rows were fetched at a time. It may indicate the level to which array fetch capabilities have been utilized. A ratio close to one may indicate an opportunity to optimize code by using array fetches. |

*Table 4–1 Predefined Data Views Provided By Oracle Trace (Cont.)*

| View Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Sorts on Disk** | Queries that did the greatest number of sorts on disk. | Sort statistics for SQL statements.<br>Number of in memory sorts.<br>Total number of rows sorted. | Sorts on disk are sorts that could not be performed in memory, therefore they are more expensive since memory access is much faster than disk access. |
| **Sorts in Memory** | Queries that did the greatest number of sorts in memory. | Sort statistics for SQL statements.<br>Number of disk sorts.<br>Total number of rows sorted. | Sorts in memory are sorts that could be performed completely within the sort buffer in memory without using the temporary tablespace segments. |
| **Rows Sorted** | Queries that sorted the greatest number of rows. | Number of in memory sorts.<br>Number of sorts on disk. | Returns sort statistics for SQL statements ordered by queries that sorted the greatest number of rows. |
| **Waits by Total Wait Time** | Highest total wait time per distinct type of wait. | Average wait time, total wait time and number of waits per wait type. | Waits are sorted by wait description or type that had the greatest cumulative wait time for all occurrences of the wait type within the collection. |
| **Waits by Average Wait Time** | Highest average wait time per wait type. | Average wait time, total wait time and number of waits per wait type. | Waits are sorted by wait type that had the greatest average wait time for all occurrences of the wait type within the collection. |
| **Waits by Event Frequency** | Frequency of waits per wait type. | Number of waits per wait type, average wait time, and total wait time. | Waits are sorted by wait events or wait descriptions that appear most frequently within the collection. |

## Accessing the Definition of a Data View

To access the definition for each data view, click on the data view name listed in the Data View Name column, for example, Disk Reads/Logical Reads Ratio. The description of the statistical data returned from the data view is on the General property page located on the right pane of the screen as shown in Figure 4–5. More details are shown in the Items and Options pages.

- General

  Displays the name, short description, and full description of the data view. For the Oracle Server predefined data views, the name reflects the name of the event. The full description is usually used to indicate what type of information is displayed in a data view and when it would be appropriate to use the data view.

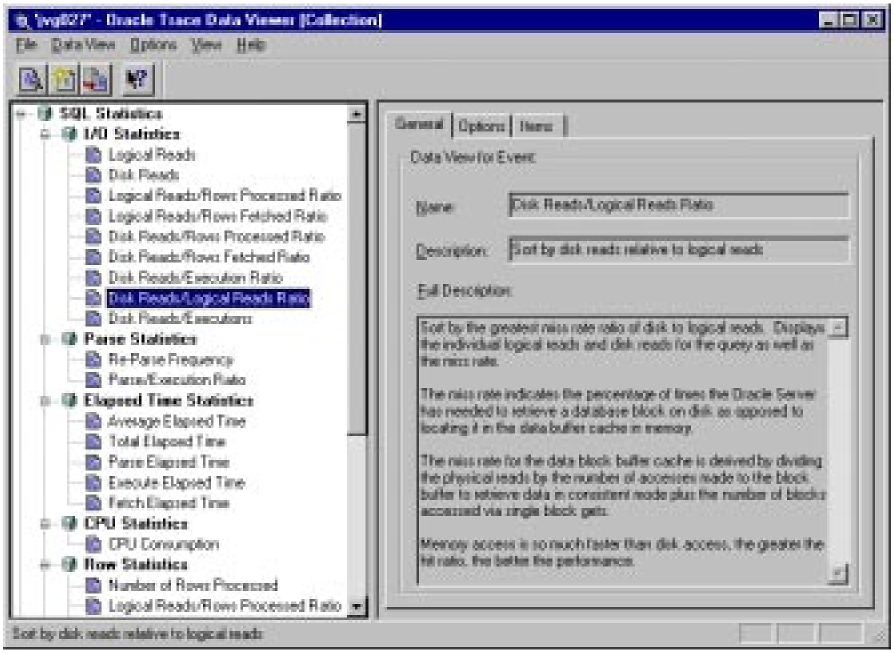
- Options

Shows which statistics will be used to determine sort order, if any. Also displays the number of rows that will be selected by the data view. Limiting the number of rows to be retrieved and displayed can accelerate the display of the data.

- Items

  Lists the statistics that will be displayed by the currently selected data view, as well as all possible statistics for the data view.

*Figure 4–5 Predefined Data View - Disk Reads/Logical Reads Ratio*



Data View for Event

Name: Disk Reads/Logical Reads Ratio

Description: Sort by disk reads relative to logical reads

Full Description:

Sort by the greatest miss rate ratio of disk to logical reads. Displays the individual logical reads and disk reads for the query as well as the miss rate.

The miss rate indicates the percentage of times the Oracle Server has needed to retrieve a database block on disk as opposed to locating it in the data buffer cache in memory.

The miss rate for the data block buffer cache is derived by dividing the physical reads by the number of accesses made to the block buffer to retrieve data in consistent mode plus the number of blocks accessed via single block gets.

Memory access is so much faster than disk access, the greater the hit ratio, the better the performance.

Sort by disk reads relative to logical reads

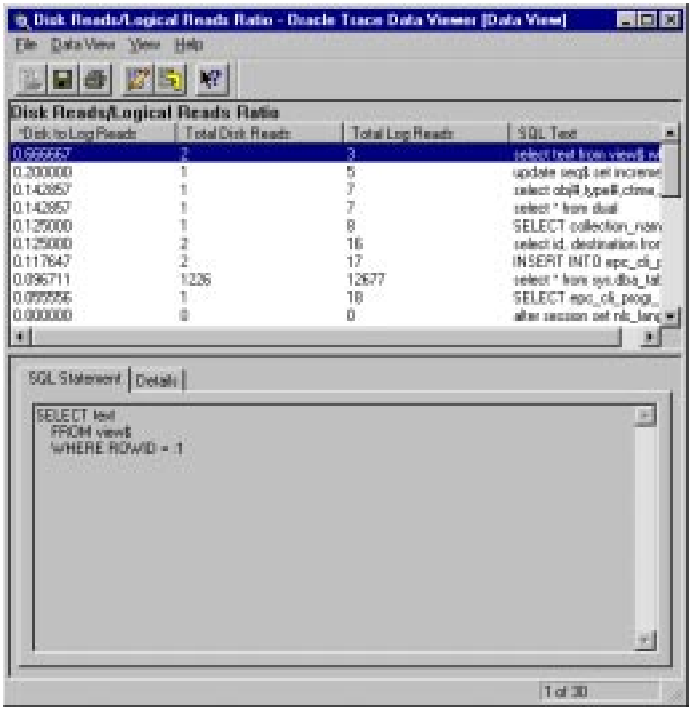
## Viewing Data in a Data View

To examine the data defined in each data view, do one of the following:

- Double-click on the View Name, for example, Disk Reads/Logical Reads Ratio.
- With the data view selected, click the Open Data View button.
- Choose Data View=>Open from the Collection window.

Data Viewer then displays the data (see Figure 4–6) identified by criteria defined in the selected data view. In this example, the data view shows summary information for each SQL statement executed, sorted by the Disk to Logical Reads Ratio (for example, column name marked with asterisk (*)).

*Figure 4–6 Details of "Disk Reads/Logical Reads Ratio"Data View*



Disk Reads/Logical Reads Ratio - Oracle Trace Data Viewer [Data View]
File
Data View
View
Help
SQL Statement
Details
1 of 30

Disk Reads/Logical Reads Ratio

| *Disk to Log Reads | Total Disk Reads | Total Log Reads | SQL Text |
|---|---|---|---|
| 0.666667 | 2 | 3 | select text from view$ w |
| 0.200000 | 1 | 5 | update seq$ set increme |
| 0.142857 | 1 | 7 | select obj#,type#,ctime,_ |
| 0.142857 | 1 | 7 | select * from dual |
| 0.125000 | 1 | 8 | SELECT collection_nam |
| 0.125000 | 2 | 16 | select id, destination fror |
| 0.117647 | 2 | 17 | INSERT INTO epc_cli_ |
| 0.096711 | 1226 | 12677 | select * from sys.dba_tal |
| 0.055556 | 1 | 18 | SELECT epc_cli_progr_ |
| 0.000000 | 0 | 0 | alter session set nls_lang |

```
SELECT text
  FROM view$
  WHERE ROWID = :1
```

The major portions of the screen are:

### Data value columns

The columns are displayed in the sort order determined by the value in the column marked with asterisk (*).

### SQL Statement

The SQL statement property page displays the SQL statement text for the query associated with the currently selected row of returned statistics. Note that statement text changes as you highlight various rows.

Use the right mouse button to display the popup menu that can be used to select and copy text.

### Details

The Details property page lists all statistics associated with the highlighted row in the data view. The statistics shown are for *all* executions of the current query within the Oracle Trace collection.

Use the right mouse button to display the popup menu that can be used to select and copy text.

---

**Note:** Detail statistics for SQL data views are similar to statistics presented by the TKPROF utility.

---

### Status Bar

Shows the number of rows returned and the number of the currently selected row. The status is displayed in the lower right corner of the window.

## Creating Custom Data Views

You can also create a new data view or create a data view like an existing data view. Create a data view when a predefined data view does not contain the information you want to examine. "Using the Data View Wizard" on page 4-19 describes the process of creating a data view.

## Modifying or Extending Data Views

You can modify a data view to either add or remove statistics from the view.

For example, to add the "Execute rows" statistics to the data view, do the following:

1. Select a data view, for example, "Disk Reads/Logical Reads Ratio," then in the Data View window, choose Data View=>Modify.
2. In the Edit Data View dialog box, select the Items tab, and move the "Execute rows" statistics from the Items list to the Display These Items list.
3. Click OK.

The new information will be added to the view immediately. You can then save this data view to use at another time by choosing File=>Save As. The original view still exists. The modified view appears at the bottom of the navigator tree under the Custom folder.

Note that from the Data View window you can:

- Click on the current sort column to reverse the sort order. Click on a new column to change the sort criteria. The sort column is indicated by an asterisk (*) appended to the front of the header title.

  Changing the sort order causes the database to be queried again for the top rows using the new sort criteria.

- Revise the options, for example, the maximum number for rows retrieved. Click the Options tab to make these changes. (This tab becomes available after you select Data View=>Modify.)
- Modify the name, description, and full description of the data view by clicking the General tab. (This tab becomes available after you select Data View=>Modify.)

## Drill-Down of Data Views

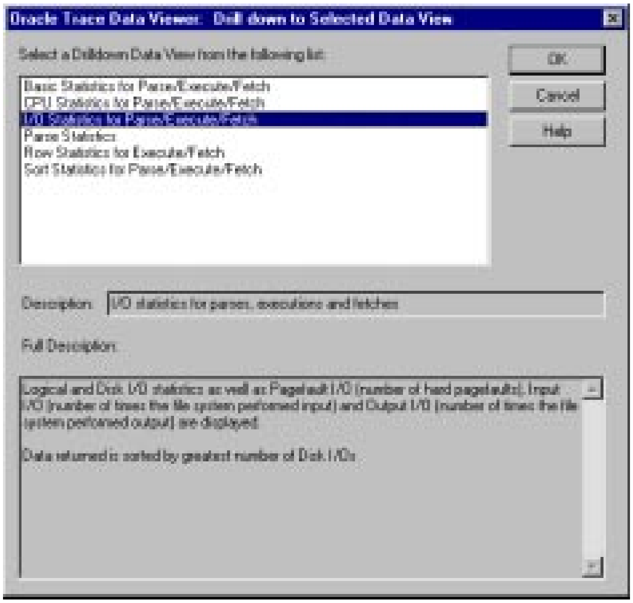
For certain events, you can drill down to related events to refine your view of the data. The drill-down option is available if the event that is currently being shown consists of other related events. For example, when examining the SQL Statistics data views, you can drill down to the Parse, Execute, and Fetch statistics for the currently selected query.

**Note:** When viewing SQL Statistics, you are viewing statistics for ALL executions of each distinct query within the collection. Drill down to view statistics for each parse, execution, and fetch for the selected SQL statement.

Select a row that you wish to analyze further (see Figure 4–6) and choose Data View=>Drill down. This allows you to join this row to other events and items of interest.

Figure 4–7 shows the resulting drill-down window.

*Figure 4–7   Intermediate Drill-Down to Selected Data View*



Select a Drilldown Data View from the following list:

Basic Statistics for Parse/Execute/Fetch
CPU Statistics for Parse/Execute/Fetch
I/O Statistics for Parse/Execute/Fetch
Parse Statistics
Row Statistics for Execute/Fetch
Sort Statistics for Parse/Execute/Fetch

Description: I/O statistics for parses, executions and fetches

Full Description:

Logical and Disk I/O statistics as well as Pagefault I/O (number of hard pagefaults), Input I/O (number of times the file system performed input) and Output I/O (number of times the file system performed output) are displayed.

Data returned is sorted by greatest number of Disk I/Os

The major portions of the window are:

- List of predefined data views

  Predefined data views are supplied for drilling down to the individual parses, executions, and fetches associated with a selected SQL statement.

- Description

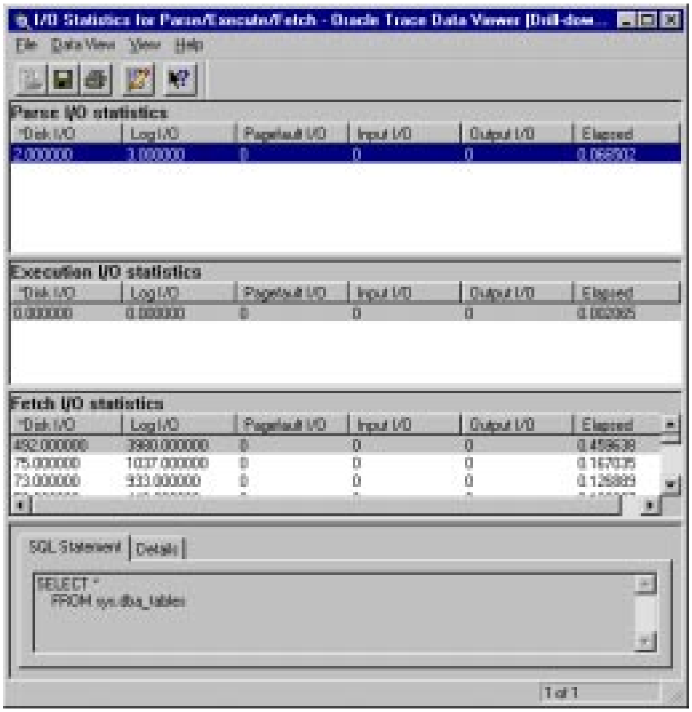
  Displays a short description of the currently selected drill-down data view.

- Full Description

  Displays a more in-depth description of the data returned by the currently selected drill-down data view, as well as a list of the statistics returned by this data view and the statistic that is used to sort it.

Select and double-click a drill-down data view to see individual parse, execute, and fetch statistics for each execution of the SQL statement within the collection. You will see the parse, execution, and fetch statistics within their own scrolling regions, as shown in Figure 4–8.

*Figure 4–8    Resulting Drill-Down Data View*


I/O Statistics for Parse/Execute/Fetch - Oracle Trace Data Viewer [Drill-dow...
File
Data View
View
Help
SQL Statement
Details
1 of 1

Parse I/O statistics

| *Disk I/O | Log I/O | Pagefault I/O | Input I/O | Output I/O | Elapsed |
|---|---|---|---|---|---|
| 2.000000 | 3.000000 | 0 | 0 | 0 | 0.068902 |

Execution I/O statistics

| *Disk I/O | Log I/O | Pagefault I/O | Input I/O | Output I/O | Elapsed |
|---|---|---|---|---|---|
| 0.000000 | 0.000000 | 0 | 0 | 0 | 0.002065 |

Fetch I/O statistics

| *Disk I/O | Log I/O | Pagefault I/O | Input I/O | Output I/O | Elapsed |
|---|---|---|---|---|---|
| 492.000000 | 3980.000000 | 0 | 0 | 0 | 0.459638 |
| 75.000000 | 1037.000000 | 0 | 0 | 0 | 0.167035 |
| 73.000000 | 933.000000 | 0 | 0 | 0 | 0.126889 |

```
SELECT *
  FROM sys.dba_tables
```

Major portions of the screen are:

- Top part of window

  Lists the detailed information returned by the data view. Each event will have columns containing statitics of items returned by the data view.

- SQL Statement

  The SQL statement property page displays the SQL statement text for the query associated with the currently selected row of returned statistics.

- Details

  The Details property page shows the statistics for *all* executions of the current query within the Oracle Trace collection.

## Selecting a Drill-Down Data View

Predefined data views are supplied for drilling down to the individual parses, executions, and fetches associated with a selected SQL statement. Table 4–2 lists the drill-down data views available in Data Viewer.

*Table 4–2 Drill-Down Data Views*

| Drill-Down Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Basic Statistics for Parse/Execute/Fetch** | Greatest elapsed time. | For each distinct event: CPU time.<br>Elapsed time.<br>Disk I/O.<br>Logical I/O.<br>Number of rows processed. | Parse, Execution, and Fetch statistics like statistics output from TKPROF. |
| **CPU Statistics for Parse/Execute/Fetch** | Greatest elapsed CPU time. | CPU total.<br>Pagefaults. | CPU and pagefault statistics for Parses, Executions, and Fetches of the current query.<br>CPU total is the number of clock ticks in both user and system mode. The clock tick granularity is specific to the operating system on which the database resides. |
| **I/O Statistics for Parse/Execute/Fetch** | Greatest number of disk I/Os. | Logical and Disk I/O statistics.<br>Pagefault I/O (number of hard pagefaults).<br>Input I/O (number of times the file system performed input).<br>Output I/O (number of times the file system performed output). | I/O statistics for parses, executions, and fetches. |
| **Parse Statistics** | Greatest elapsed time. | Current user identifier.<br>Schema identifiers. | Parse information, such as whether or not the current statement was missed in library cache, Oracle optimizer mode, current user identifier, and schema identifier. |
| **Row Statistics for Execute/Fetch** | Greatest number of rows returned. | Number of rows returned.<br>Number of rows sorted.<br>Number of rows returned during a full table scan. | Execution and fetch row statistics. |

*Table 4–2 Drill-Down Data Views (Cont.)*

| Drill-Down Name | Sort By | Data Displayed | Description |
|---|---|---|---|
| **Sort Statistics for Parse/Execute/Fetch** | Greatest elapsed time. | Sorts on disk.<br>Sorts in memory.<br>Number of rows sorted.<br>Number of rows returned from a full table scan. | Parse, execution, and fetch sort statistics. |
| **Wait Parameters** | Wait_Time. | Description.<br>Wait_Time.<br>P1.<br>P2.<br>P3. | Investigating Waits may help to identify where contention resides and what is causing the contention.<br>P1, P2, and P3 parameters are values that provide more information about a specific wait event. The parameters are foreign keys to views that are wait event dependent. For example, for latch waits, P2 is the latch number which is a foreign key to v$latch.<br>The meaning of each parameter is specific to each wait type. |

## Modifying Drill-Down Data Views

Drill-down data views can be modified and saved much like top level data views. You can modify a drill-down data view to either add or remove statistics from the view.

Because slightly different statistics are associated with the parse, execution, or fetch events, you must modify each event separately. For example, to add the "Rows" (row count) statistic to the execution and fetch output in the drill-down data view, do the following:

1. Select any row in "Execution I/O Statistics" and choose Data View=>Modify.
2. In the Edit Data View dialog box, select the Items tab, and move the Rows statistics from the Items list to the Display These Items list.
3. Click OK.
4. Select any row in the "Fetch I/O Statistics" and add the "Rows" statistics to that event.

The new information is added to the view immediately. You can then save this data view to use at another time.

## Saving As Data Views and Files

From the Data View window, you can save the current data view settings by selecting File=>Save As=>Data View. This is useful if you want to reuse this data view frequently. You must provide a name, description, and full description to save the view. Modified views are listed in the Custom folder in the navigator tree.

Also, you can save the selected rows of a data view to a file by selecting File=>Save As=>File. You can save the data in either of the following formats:

- SQL text (text only) for rows that have SQL text. You can use this file for import into SQL Analyze.
- CSV (comma-separated value format). You can use this format for import into other programs that accept CSV format.

## Printing Data Views

From the File menu, there are three printing options available: Print, Print Setup, and Set Printer Font.

The Print option prints the currently active portion of a data view. If the currently active region is a data area, you can print either the entire region (including rows that are not visible on the screen) or selected rows. If the currently active region is the "SQL Statement / Details" area, both the SQL Statement and the Details will be printed.

The Print Setup option displays the standard Print Setup dialog box.

The Set Printer Font option displays the standard Font dialog box with the font selection for the currently selected printer.

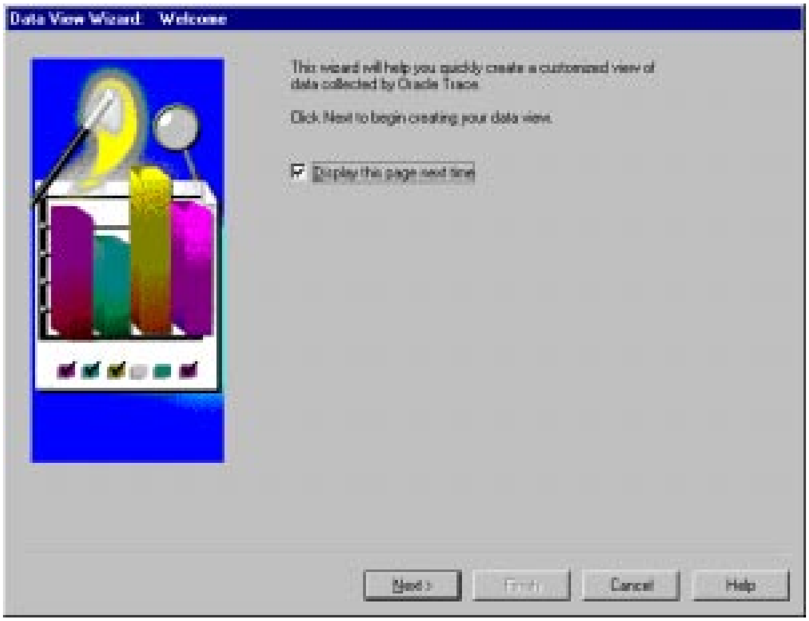
# Using the Data View Wizard

You can create your own customized data view by using the Oracle Trace Data View wizard (hereafter referred to as the Data View wizard).

To invoke the wizard, choose Data View=>Create or Data View=>Create Like. The wizard’s Welcome page appears (see Figure 4–9).

Use the Data View wizard to examine data that you cannot find in a predefined view. However, this data will not be optimized and may take longer to process.

*Figure 4–9 Data View Wizard Welcome Page*



Data View Wizard: Welcome

This wizard will help you quickly create a customized view of data collected by Oracle Trace.

Click Next to begin creating your data view.

☑ Display this page next time

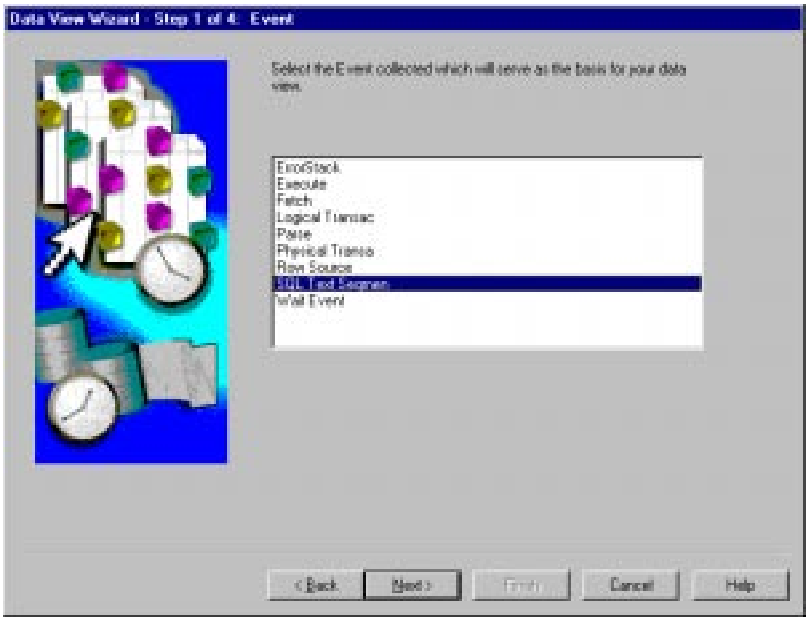
You can choose to bypass this Welcome page the next time you use the Data View wizard.

Click Next to start the definition of the data view.

## Choosing an Event

The Event page is the first page used in the definition of the data view that you are creating. The Event page is as shown in Figure 4–10.

*Figure 4–10 Selecting the Event*



Data View Wizard - Step 1 of 4: Event

Select the Event collected which will serve as the basis for your data view.

ErrorStack
Execute
Fetch
Logical Transac
Parse
Physical Transa
Row Source
SQL Text Segmen
Wait Event

From the Event page, choose the event on which to base the data view. The list of event names comes from the Oracle Trace formatted data tables. Event names are limited to and truncated at 16 characters.

If you are looking at an Oracle Server collection and you let Data Viewer compute statistics, the SQL Text Segmen event allows you to examine statistics from Parse, Execute, and Fetch events.

After you select the event, continue to the Items page by clicking Next.

---

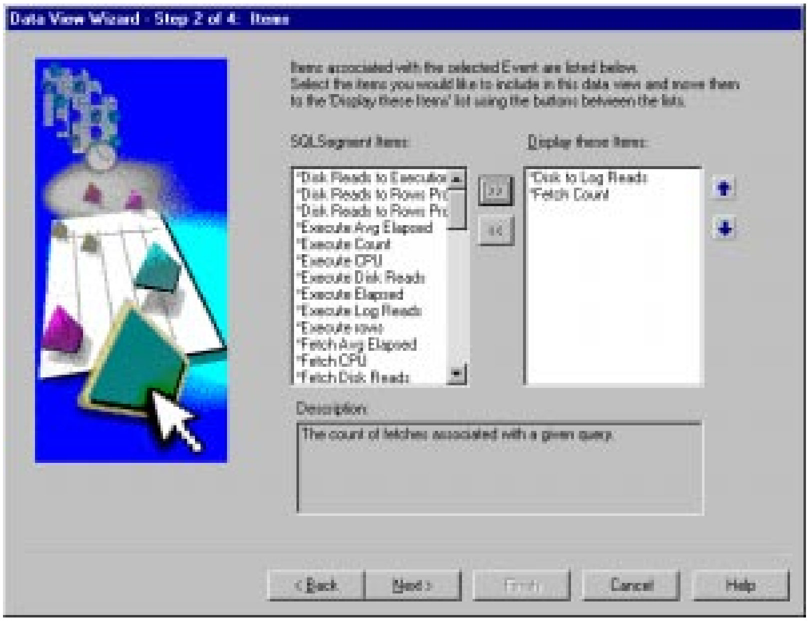
**Note:** For information about events and items refer to "Collecting Event Data" on page 1-3.

---

## Choosing Items Associated with the Selected Event

On the Items page, select the items that you want to display in the data view (see Figure 4–11).

*Figure 4–11 Selecting and Ordering Items Associated with the Event*



Data View Wizard - Step 2 of 4: Items

Items associated with the selected Event are listed below.
Select the items you would like to include in this data view and move them to the 'Display these Items' list using the buttons between the lists.

SQLSegment Items:

*Disk Reads to Executio
*Disk Reads to Rows Pr
*Disk Reads to Rows Pr
*Execute Avg Elapsed
*Execute Count
*Execute CPU
*Execute Disk Reads
*Execute Elapsed
*Execute Log Reads
*Execute rows
*Fetch Avg Elapsed
*Fetch CPU
*Fetch Disk Reads

Display these Items:

*Disk to Log Reads
*Fetch Count

Description:

The count of fetches associated with a given query.

Items available are based on the event chosen.

- Choose items to display by selecting them in the Items list and use >> to move the items to the Display These Items list. Use << to remove an item from the Display These Items list. Items prefixed with an asterisk (*) are statistics computed by Data Viewer.
- Order these items by highlighting an item in the Display These Items list and using the up and down arrows to move the item in the list. The first item in the list is the first item displayed in the data view. Each item will be in a column in the data view.

When the item is selected, the Description field displays the description.

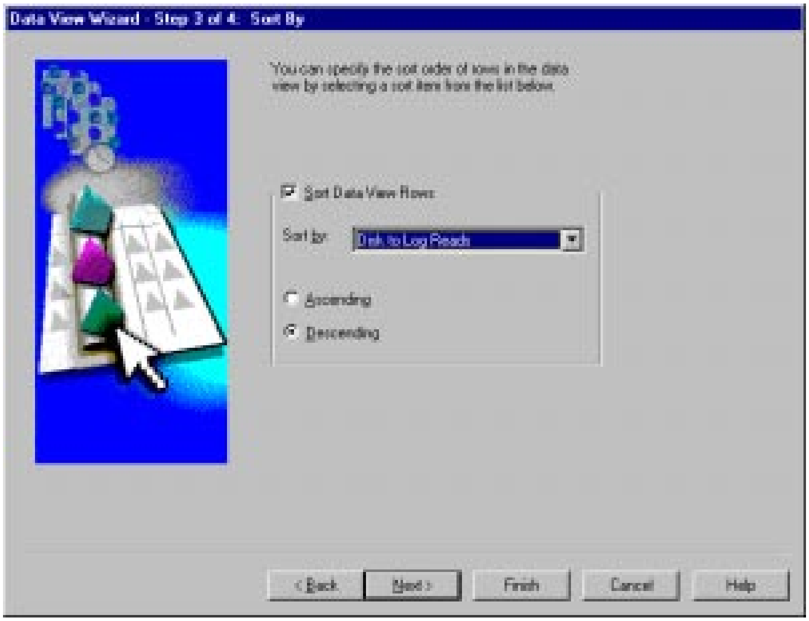
You must choose at least one item to generate a data view.

The next step in defining the data view is to define the sort criteria. Click Next to access the Sort By page.

## Choosing the Sort Criteria

The Sort By page (see Figure 4–12) allows you to choose the item or statistic on which the sorting will take place. Sort criteria is optional.

*Figure 4–12 Selecting Sort Criteria*



Data View Wizard - Step 3 of 4: Sort By

You can specify the sort order of rows in the data view by selecting a sort item from the list below.

☑ Sort Data View Rows

Sort by: Disk to Log Reads

○ Ascending

● Descending

The data can be displayed in either ascending or descending order. Descending order is the default.

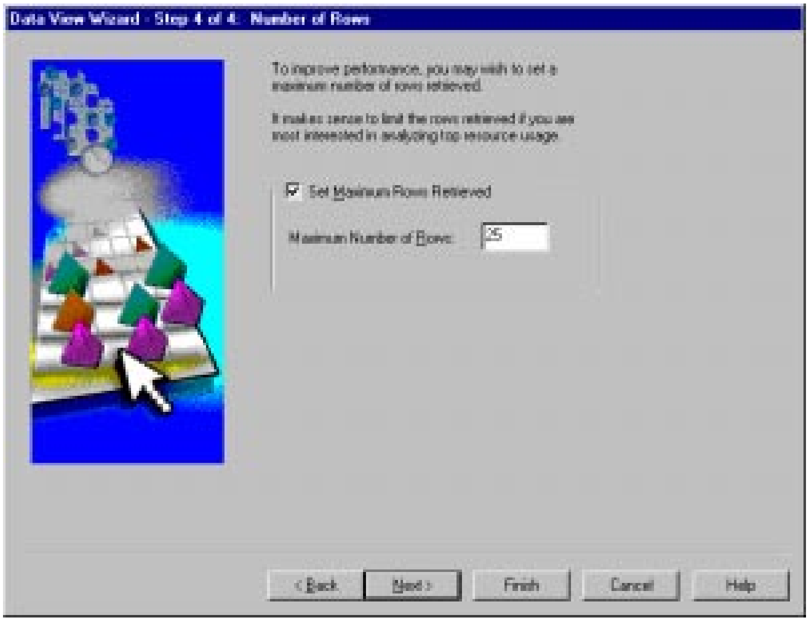
The next step in defining the data view is to optionally limit the number of rows to display in the data view. Click Next to access the Number of Rows page.

## Defining the Number of Rows to Display

The last step in defining the data view is to decide how many rows are to be displayed in the data view (see Figure 4–13). You can either limit the number of rows to display or decide to display all the rows.

Limiting the number of rows returned can improve performance. If you are interested in viewing the highest value of some statistic, limit the number of rows. You can change the number for rows from the data view later on.

*Figure 4–13 Limiting Number of Rows*



Data View Wizard - Step 4 of 4: Number of Rows

To improve performance, you may wish to set a maximum number of rows retrieved.

It makes sense to limit the rows retrieved if you are most interested in analyzing top resource usage.

☑ Set Maximum Rows Retrieved

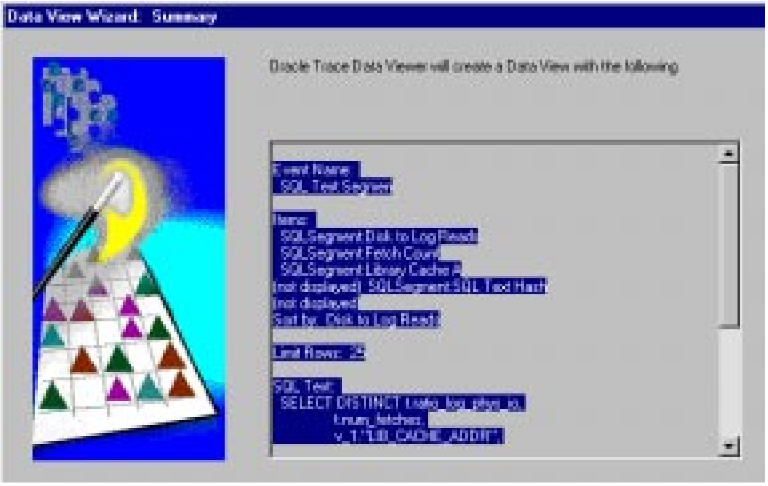
Maximum Number of Rows: 25

This is the last step in defining the data view. Click Next to view the Summary page.

## Reviewing the Summary Page

The Summary page gives you the opportunity to review the choices that you made while defining the data view (see Figure 4–14).

*Figure 4–14 Summary Page*



Oracle Trace Data Viewer will create a Data View with the following

```
Event Name:
  SQL Text Segment

Items:
  SQLSegment Disk to Log Reads
  SQLSegment Fetch Count
  SQLSegment Library Cache A
(not displayed) SQLSegment SQL Text Hash
(not displayed
Sort by: Disk to Log Reads

Limit Rows: 25

SQL Text:
  SELECT DISTINCT t.ratio_log_phys_io,
           t.num_fetches,
           v_1."LIB_CACHE_ADDR",
```

You can either accept all of the choices by clicking Finish or go back to change one or more definitions by clicking Back.

# A

# Manual Collection of Oracle Trace Data

Though Oracle Trace Manager (graphical user interface) is the primary interface to Oracle Trace, optionally you can perform a manual collection of Oracle Trace data. This can be done by using a command-line interface, using initialization parameters, or executing stored procedures.

## Using the Oracle Trace Command-Line Interface

Another option for controlling Oracle Trace server collections is the Oracle Trace CLI (Command-line Interface). The CLI is invoked by the OTRCCOL command for the following functions:

- OTRCCOL START job_id input_parameter_file
- OTRCCOL CHECK col_name
- OTRCCOL STOP job_id input_parameter_file
- OTRCCOL FORMAT input_parameter_file
- OTRCCOL DCF col_name cdf_file (to delete collection files)
- OTRCCOL DFD col_name username password service col_id (to delete formatted data from the database)

The parameter JOB_ID can be any numeric value. The input parameter file contains specific parameter values required for each function as shown in the following examples. COL_NAME (collection name) and CDF_FILE (collection definition file) are initially defined in the START function input parameter file.

The OTRCCOL START command invokes a collection based upon parameter values contained in the input parameter file. For example:

```
otrccol  start  1234  my_start_input_file
```

Where my_start_input_file contains the following input parameters:

```
col_name= my_collection
dat_file= <usually same as collection name>.dat
cdf_file= <usually same as collection name>.cdf
fdf_file= <server event set>.fdf
regid= 1   192216243   0   0   5   <database SID> [Use the database SID for an
Oracle8 database collection or service name if this is an Oracle7 database
collection.]
```

> **Note:** White space after the equal sign (=) is **required** for Oracle Trace release 7.3.x command-line interface.

These parameter file records are all required for database collection and the regid record must contain the values shown including your SID or service name value. The server event sets that can be used as values for the fdf_file are **oracle**, **oraclec**, **oracled**, **oraclee,** and **oraclesm**. See "Using Initialization Parameters to Control Oracle Trace" on page A-3 for more information on the server event sets.

The OTRCCOL STOP command halts a running collection as follows:

```
otrccol stop 1234 my_stop_input_file
```

Where my_stop_input_file contains the collection name and cdf_file name. You can use the same database input parameter file as used with the START command.

The OTRCCOL FORMAT command formats the binary collection file to Oracle tables. An example of the FORMAT command is:

```
otrccol format my_format_input_file
```

Where my_format_input_file contains the following input parameters

```
username=  <database username>
password=  <database password>
service=  <database service name>
cdf_file= <usually same as collection name>.cdf
full_format= <0/1>
```

A full_format value of 1 produces a full format; a value of 0 produces a partial format. (You may find it easier to use the Oracle Trace formatter image, otrcfmt, rather than the CLI format.) See "Formatting Oracle Trace Data to Oracle Tables" on page A-7 for information on formatting part or all of an Oracle Trace collection and for other important information on creating the Oracle Trace formatting tables prior to running the format command.

The OTRCCOL DCF command deletes collection files for a specific collection. The OTRCCOL DFD command deletes formatted data from the Oracle Trace formatter tables for a specific collection.

## Using Initialization Parameters to Control Oracle Trace

There are six Oracle Trace database initialization parameters that are set up by default to control Oracle Trace. By logging in to the internal (or other privileged) account in your database and executing a “show parameters trace” command, you will see the following parameters:

| NAME | TYPE | VALUE (Oracle7 Databases) | VALUE (Oracle8 Databases) |
|---|---|---|---|
| ORACLE_TRACE_COLLECTION_NAME | string | oracle | NULL |
| ORACLE_TRACE_COLLECTION_PATH | string | ?/rdbms/log | ?/otrace/admin/cdf |
| ORACLE_TRACE_COLLECTION_SIZE | integer | 5242880 | 5242880 |
| ORACLE_TRACE_ENABLE | Boolean | FALSE | FALSE |
| ORACLE_TRACE_FACILITY_NAME | string | oracle | oracled |
| ORACLE_TRACE_FACILITY_PATH | string | ?/rdbms/admin | ?/otrace/admin/fdf |

The Oracle Trace parameters may be modified and used by adding them to your INIT*sid*.ORA file.  To start tracing for a database using these parameters, you must minimally add (or uncomment) the ORACLE_TRACE_ENABLE=TRUE parameter to your .ORA file. You also need to provide a non-null value to the ORACLE_ TRACE_COLLECTION_NAME parameter (which is null for Oracle8). These parameters are in addition to default parameter values which are provided.

### Enabling Oracle Trace Collections for Oracle8 Server

Note that the ORACLE_TRACE_ENABLE parameter is set to FALSE by default. A value of FALSE disables any use of Oracle Trace for the Oracle8 server. (For Oracle7, the parameter only needs to be TRUE if you are using INIT.ORA to start an Oracle Trace database collection.)

To enable Oracle Trace collections for the server, the parameter is set to TRUE. Having the parameter set to TRUE does not start an Oracle Trace collection, but allows Oracle Trace to be used for that server. Oracle Trace can then be started in one of the following ways:

- Using the Oracle Trace Manager application.
- Setting the ORACLE_TRACE_COLLECTION_NAME parameter. (For Oracle Server release 8.0.4, you need to set the collection name to get a collection on database startup.)

The default value for this parameter is NULL. A collection name can be up to 16 characters in length. You must then shut down your database and start it up again to make the parameters take effect. If a collection name is specified, when you start the server you automatically start an Oracle Trace collection for all database sessions.

To stop the collection that was started using the ORACLE_TRACE_COLLECTION_NAME parameter, shut down the server instance and reset the ORACLE_TRACE_COLLECTION_NAME to NULL. The collection name specified in this value is also used in two collection output file names: the collection definition file (*collection_name*.cdf) and the binary data file (*collection_name*.dat).

## Determining the Event Set that Oracle Trace Collects

The ORACLE_TRACE_FACILITY_NAME determines the event set that Oracle Trace collects.

Possible values for the ORACLE_TRACE_FACILITY_NAME parameter are:

- **oracle** for the server ALL event set (includes wait events)
- **oraclec** for the CACHEIO event set (caching statistics for buffer cache I/O)
- **oracled** for the server DEFAULT event set
- **oraclee** for the EXPERT event set (Oracle Expert application)
- **oraclesm** for the SUMMARY event set (workload statistics for Summary Advisor application)

You must then shut down your database and start it up again before the parameters take effect.  If you chose the ALL event set **oracle**, you may want to set TIMED_STATISTICS=TRUE in your INIT.ORA file to enable collection of wait times.

Once restarted, the database begins collecting data for the class of data that you selected. To stop the collection, you must shut down the database, set the INIT*sid*.ORA parameter ORACLE_TRACE_ENABLE=FALSE, and restart the database.

If, once restarted, the database does not start collecting data, you should check the following:

- The event set file, identified by the oracle_trace_facility_name (with .fdf appended to it) is located in the directory identified by oracle_trace_facility_path.
- The following files exist in your $ORACLE_HOME/otrace/admin directory: regid.dat, collect.dat, and facility.dat (or product.dat for Oracle7.x databases). If they do not, you must run the otrccref executable, which is located in $ORACLE_HOME/bin, to create them.
- The Oracle Trace parameters are set to the values that you changed in the INIT*sid*.ORA file. There is a 35-character maximum to the strings, so always try to use the ?/ to indicate $ORACLE_HOME instead of using the full path.
- Look for an EPC_ERROR.LOG file. It will give you more information about why a collection may have failed.
- Look for *.trc files in the USER_DUMP_DEST directory. Searching for epc in the *.trc files may provide errors. These errors and their descriptions are located in the $ORACLE_HOME/otrace/include/epc.h file.

## Using Stored Procedure Packages to Control Oracle Trace

You can invoke an Oracle Trace collection for your own session or for another session by using the Oracle Trace stored procedure packages. This method only works for database sessions that are currently active.

To collect Oracle Trace data for your own database session, execute the following stored procedure package:

```
dbms_oracle_trace_user.set_oracle_trace(true/false,
collection_name, server_event_set)
```

- true/false = BOOLEAN: TRUE to turn on, FALSE to turn off
- collection_name = varchar2: collection name (no file extension, 8-character maximum)
- server_event_set = varchar2: server event set (oracled, oracle, or oraclee)

Example:

```
EXECUTE dbms_oracle_trace_user.set_oracle_trace(TRUE,'MYCOLL','oracle');
```

To collect Oracle Trace data for a database session other than your own, execute the following stored procedure package:

```
dbms_oracle_trace_agent.set_oracle_trace_in_session(sid, serial#,
true/false, collection_name, server_event_set)
```

- sid = number:  session instance from v$session.sid
- serial# = number: session serial number from v$session.serial#

Example:

```
EXECUTE dbms_oracle_trace_agent.set_oracle_trace_in_session
(8,12,TRUE,'NEWCOLL','oracled');
```

If the collection does not occur, you should check the following:

- The server event set file identified by server_event_set exists.  If you do not include a full file specification in this field, then the file should be located in the directory identified by oracle_trace_facility_path in the INIT*sid*.ORA file.
- The following files exist in your $ORACLE_HOME/otrace/admin directory: regid.dat, facility.dat, and collect.dat.  If they do not, you must run the otrccref executable to create them.
- The stored procedure packages exist in the database.  If the packages do not exist, then you must run the $ORACLE_HOME/otrace/admin/otrcsvr.sql file to create the packages. Execute this SQL script while logged into the database SYS or Internal account.
- The user has the EXECUTE privilege on the stored procedure.

## Oracle Trace Collection Results

Running an Oracle Trace collection produces the following collection files located in the directory specified in INIT.ORA or in the default collection path:

- *collection_name*.cdf is the Oracle Trace collection definition file for your collection.
- *collection_name*.dat files are the Oracle Trace output files containing the data in binary format.

You can access the Oracle Trace data contained in the collection files in the following ways:

- Oracle Trace Data Viewer (data must be formatted)

- Format the data to the tables in the Oracle database for SQL access and reporting.
- Create Oracle Trace reports from the binary file.

## Formatting Oracle Trace Data to Oracle Tables

Your Oracle Trace server collection can be formatted to Oracle tables for more flexible access by any SQL reporting tool. This section describes the command-line interface for formatting.

Oracle Trace produces a separate table for each event collected. For example, a “parse” event table is created to store data for all parse events that occur during a server collection. If the version of Oracle Trace is prior to 7.3.4 or 8.0.4, then before you can format data, you must first set up the Oracle Trace formatter tables by executing the OTRCFMTC.SQL script on the server host machine while logged into the database as the user who will own the formatted data.

Use the following command to format an Oracle Trace collection:

```
OTRCFMT [-f] [-c#] collection_name.cdf [user/password@database]
```

If user/password@database is omitted, you will be prompted for this information.

The optional formatting parameters are defined as follows:

**-f**
Formats the entire data file, regardless of whether or not portions of the file have been formatted at a previous time. This is useful if you have data that was previously formatted to another database and you want to format it to a new database.

If you omit the -f parameter from the command line, only data that has not previously been formatted is processed. This is useful when you want to format data for a collection while that collection is still in progress. You can later format only new data for the collection by omitting the -f parameter from the command line.

**-c#**
Specifies the commit interval (the number of inserts into the database performed before a commit is made). If you omit the -c parameter, a system default is used.

#### user/password@database

Specifies the user name, password, and database. If you do not enter this information on the command line, you are prompted for it. Descriptions of these items follow.

#### Username

Enter the Oracle user name under which the Oracle Trace formatter tables were created, or enter the owner of a schema containing formatter tables.

#### Password

Enter the password for the user name.

#### Database

Enter a service name for the database. Do not enter a system identifier (SID). If you enter nothing, Oracle Trace will use the default database.

## New otrcfmt Image for Oracle Server Releases 7.3.4 and 8.0.4

The otrcfmt image for Oracle Server releases 7.3.4 and 8.0.4 adds two new columns to the event data tables: elapsed_secs and elapsed_nano_secs. These columns are added to all duration events.

Because of the new columns, there is a compatibility issue using older otrcfmt images to format to the same database user.

The 7.3.4 and 8.0.4 otrcfmt image can format data into a database that contains previously formatted data from any prior version of Oracle Trace. However, when a table has been updated by the 7.3.4 and 8.0.4 otrcfmt image, it can no longer be formatted by an 8.0.3, 7.3.3, or prior otrcfmt image.

The otrcfmt image executes on the collection target database. The version of the image depends on the collection target database, not the format target database. If you are collecting data for databases with different versions, you will be using multiple versions of the otrcfmt image.

If you plan to have multiple versions of the otrcfmt image directing output to the same database, use separate database user accounts to differentiate the formatted data. For example, you could create user_803 and user_804 to keep the data separate.

If you do have this conflict but you do not need to keep the formatted data, delete the old formatted data tables using otrcfmtd.sql and then re-create the format tables using the otrcfmtc.sql script located in the otrace/admin directory. If subsequent formatting will be done using the 7.3.4, 8.0.4 or later otrcfmt image, the image will create the format tables for you and you do not need to execute the otrcfmtc.sql script.

## Oracle Trace Statistics Reporting Utility

The Oracle Trace statistics reporting utility displays statistics for all items associated with each occurrence of a server event. These reports can be quite large. You can control the report output by using command parameters. Use the following command and optional parameters to produce a report:

```
otrcrep [optional parameters]  collection_name.CDF
```

The first step you may want to take is to generate a report called “collection_ PROCESS.txt.” You can produce this report first to give you a listing of specific process identifiers for which you want to run other reports.

You can manipulate the output of the Oracle Trace reporting utility by using the following optional report parameters:

| | |
|---|---|
| output_path | Specifies a full output path for the report files. If not specified, the files will be placed in the current directory. |
| -p [<pid>] | Organizes event data by process. If you specify a process ID (pid), you will have one file with all the events generated by that process in chronological order. If you omit the process ID, you will have one file for each process that participated in the collection. The output files are named *collection*_P*pid*.txt. |
| -P | Produces a report called *collection*_PROCESS.txt that lists all processes that participated in the collection. It does not include event data. You could produce this report first to determine the specific processes for which you might want to produce more detailed reports. |
| -w# | Sets report width, such as -w132. The default is 80 characters. |
| -l# | Sets the number of report lines per page. The default is 63 lines per page. |
| -h | Suppresses all event and item report headers, producing a shorter report. |
| -s | Used with Net8 data only (or SQL*Net for Oracle7). |

-a

Creates a report containing all the events for all products, in the order they occur in the data collection (.dat) file.

# B

# Oracle Server Events

The following sections describe events that have been instrumented in Oracle Server. Most of the events are useful for performance analysis and tuning and workload analysis by Oracle Expert.

There are two types of events: **point** events and **duration** events. Point events represent an instantaneous occurrence of something in the instrumented product. An example of a point event is an error occurrence. Duration events have a beginning and ending. An example of a duration event is a transaction. Duration events can have other events occur within them; for example, the occurrence of an error within a transaction.

Table B–1 lists the Oracle Server events instrumented for Oracle Trace. For more detailed descriptions, refer to the section for the event in which you are interested.

*Table B–1 Oracle Server Events*

| Event | Description | Type of Event |
|---|---|---|
| Connection | Connection to a database. | Point |
| Disconnect | Disconnection from a database. | Point |
| ErrorStack | Code stack for core dump. | Point |
| Migration | Session migration between shared server processes. | Point |
| ApplReg | Application context information. | Point |
| RowSource | Row information. For Oracle Server release 8.0.2 and higher, this also includes data about the execution plan. | Point |
| SQLSegment | Text of SQL statement. | Point |
| Wait | Generic WAIT event. Context is provided in the event strings. | Point |

*Table B–1 Oracle Server Events (Cont.)*

| Event | Description | Type of Event |
|---|---|---|
| Parse | SQL parsing information. | Duration |
| Execute | Information for execution of SQL. | Duration |
| Fetch | Actual row retrieval information. | Duration |
| LogicalTX | The first time a database command is performed that may change the database status. | Duration |
| PhysicalTX | Event marking a definite change in database status. | Duration |

# Data Items Collected for Events

Specific kinds of information, known as items, are associated with each event. There are three types of items:

- Resource utilization items
- Cross-product items
- Product-specific items

## Resource Utilization Items

Oracle Trace has a standard set of items, called resource utilization items, that it can collect for any instrumented application, including the Oracle Server. In addition, all duration events in the Oracle Server include items for database statistics specific to the Oracle Server.

The standard resource utilization items are described in Table B–2.

An Oracle Trace collection can be formatted into Oracle tables for access, analysis, and reporting. The last column contains the data type for data items formatted to the Oracle database.

*Table B–2 Standard Resource Utilization Items*

| Item Name | Description | Item ID | Datatype of Formatted Data |
|---|---|---|---|
| UCPU | Amount of CPU time in user mode | 129 | NUMBER |
| SCPU | Amount of CPU time in system mode | 130 | NUMBER |

*Table B–2 Standard Resource Utilization Items (Cont.)*

| Item Name | Description | Item ID | Datatype of Formatted Data |
|---|---|---|---|
| INPUT_IO | Number of times file system performed input | 131 | NUMBER |
| OUTPUT_IO | Number of times file system performed output | 132 | NUMBER |
| PAGEFAULTS | Number of hard and soft page faults | 133 | NUMBER |
| PAGEFAULT_IO | Number of hard page faults | 134 | NUMBER |
| MAXRS_SIZE | Maximum resident set size used | 135 | NUMBER |

The implementation of the item is platform specific; if the item is not implemented, the value is 0.

## Cross-Product Items

Oracle Trace provides a set of 14 items called cross-product items. These data items allow programmers to relate events for different products. For example, a business transaction may generate events in two products: an application and the database. The cross-product data items allow these disparate events to be joined for analysis of the entire business transaction.

Cross-product items are reserved for specific products or product types as described in Table B–3. If you do not use the products for which items are reserved, you may use those items for your own purposes.

*Table B–3 Cross-Product Items*

| Item Name | Layer | Description | Item ID | Datatype of Formatted Data |
|---|---|---|---|---|
| CROSS_FAC_1 | Application | Application ID. For use by high-level applications such as Oracle Financials, third-party or customer applications | 136 | NUMBER |
| CROSS_FAC_2 | Oracle Forms | Oracle Forms ID | 137 | NUMBER |

***Table B–3 Cross-Product Items (Cont.)***

| Item Name | Layer | Description | Item ID | Datatype of Formatted Data |
|---|---|---|---|---|
| CROSS_FAC_3 | Net8 | Remote node connection ID | 138 | NUMBER |
| CROSS_FAC_4 | Oracle Server | Transaction ID | 139 | NUMBER |
| CROSS_FAC_5 | Oracle Server | Hash_ID of SQL statement | 140 | NUMBER |
| CROSS_FAC_6 | Oracle Server release 8.x | User ID | 141 | NUMBER |
| CROSS_FAC_7 | n/a | Not reserved | 142 | NUMBER |
| CROSS_FAC_8 | n/a | Not reserved | 143 | NUMBER |
| CROSS_FAC_9 | n/a | Not reserved | 144 | NUMBER |
| CROSS_FAC_10 | n/a | Not reserved | 145 | NUMBER |
| CROSS_FAC_11 | n/a | Not reserved | 146 | NUMBER |
| CROSS_FAC_12 | n/a | Not reserved | 147 | NUMBER |
| CROSS_FAC_13 | n/a | Not reserved | 148 | NUMBER |
| CROSS_FAC_14 | n/a | Not reserved | 149 | NUMBER |

**Note:** In this version of Oracle Trace, the term “facility” has been changed to “product”. Therefore, the items named CROSS_FAC_*x* are cross-product items.

Cross-product item 1 (referred to as CROSS_FAC_1) will contain data only if data is supplied by an instrumented application.

Cross-product item 2 (CROSS_FAC_2) is reserved for use by a future release of Oracle Forms. Instrumented applications and Oracle Forms will pass identification data to the Oracle Server collection through these cross-product items.

Cross-product item 3 (CROSS_FAC_3) is reserved for use by Net8. Net8 supplies the connection ID to Oracle Trace through CROSS_FAC_3. CROSS_FAC_3 is the key element in coordinating client/server Oracle Trace collections. Oracle Trace uses the Net8 global connection ID as the common element to match in the merger of the client and server collection files. The global connection ID is the same for the client and the server connection. It is used as the Oracle Trace registration ID that gets logged with the CROSS_FAC_3 event collection.

Each Oracle Server event will record cross-product items 1 through 6.

## Items Specific to Oracle Server Events

The Oracle Server product definition file defines items specific to the Oracle Server. Use the item's number to locate it within the list. The formatted datatype describes how the Oracle Trace formatter defines the item when it formats data into an Oracle database.

The Oracle Server items are listed in Table B–4.

***Table B–4 Oracle Server Items***

| Item Name | Description | Item Number | Formatted Datatype |
|---|---|---|---|
| App_Action | Action name set by using the dbms_application_info.set_module procedure | 23 | VARCHAR2(255) |
| App_Module | Module name set using the dbms_ application_info.set_module procedure | 22 | VARCHAR2(255) |
| Commit_Abort | Indicates if a transaction committed or aborted | 24 | NUMBER |
| Consistent_Gets | Number of blocks retrieved in consistent mode (did not change the data and therefore did not create any locks or conflicts with other users) | 104 | NUMBER |
| CPU_Session | CPU session | 112 | NUMBER |
| Current_UID | Current user ID | 36 | NUMBER |
| Cursor_Number | Number of cursor associated with SQL statement | 25 | NUMBER |
| DB_Block_Change | Number of blocks changed | 102 | NUMBER |

*Table B–4 Oracle Server Items (Cont.)*

| Item Name | Description | Item Number | Formatted Datatype |
|---|---|---|---|
| DB_Block_Gets | Number of blocks retrieved in current mode. For large queries, this item tells how many sections of the database (logical pages) were fetched to retrieve all needed records. | 103 | NUMBER |
| Deferred_Logging | Value used by Oracle Trace internally | 14 | NUMBER |
| Depth | Recursive level at which SQL statement is processed | 32 | NUMBER |
| Description | Depends upon event in which it occurs | 43 | VARCHAR2(255) |
| Elapsed_Session | Elapsed time for the session | 113 | NUMBER |
| End_of_Fetch | Flag set if data retrieved is last data from query | 38 | NUMBER |
| Lib_Cache_Addr | Address of SQL statement in library cache | 27 | VARCHAR2(16) |
| Login_UID | Internal ID within the Oracle database that identifies the user ID for the session | 15 | NUMBER |
| Login_UName | Internal ID within the Oracle database that identifies the system account name for the session | 16 | VARCHAR2(255) |
| Missed | Flag set if SQL statement was missing in library cache | 33 | NUMBER |
| Object_ID[1] | Object ID of the row source | 46 | NUMBER |
| Operation[1] | Text of the operation | 47 | VARCHAR2(255) |
| Operation_ID[1] | Position of the operation within the execution plan for a statement | 28 | NUMBER |
| Optimizer_Mode | Oracle optimizer mode | 35 | VARCHAR2(32) |
| Oracle_Cmd_Type | Oracle command number | 34 | NUMBER |
| Oracle PID | Oracle process ID | 11 | NUMBER |

***Table B–4 Oracle Server Items (Cont.)***

| Item Name | Description | Item Number | Formatted Datatype |
|---|---|---|---|
| OS_Image | Operating system image (program name) | 42 | LONG |
| OS_Mach | Operating system host machine | 20 | VARCHAR2(255) |
| OS_Term | Operating system terminal | 19 | VARCHAR2(255) |
| OS_UName | Operating system username | 18 | VARCHAR2(255) |
| P1 | The definition of P1 depends upon the event in which it occurs. | 1 | NUMBER |
| P2 | The definition of P2 depends upon the event in which it occurs. | 2 | NUMBER |
| P3 | The definition of P3 depends upon the event in which it occurs. | 3 | NUMBER |
| P4 | The definition of P4 depends upon the event in which it occurs. | 4 | NUMBER |
| P5 | The definition of P5 depends upon the event in which it occurs. | 5 | NUMBER |
| P6 | The definition of P6 depends upon the event in which it occurs. | 6 | NUMBER |
| P7 | The definition of P7 depends upon the event in which it occurs. | 7 | NUMBER |
| P8 | The definition of P8 depends upon the event in which it occurs. | 8 | NUMBER |
| P9 | The definition of P9 depends upon the event in which it occurs. | 9 | NUMBER |
| P10 | The definition of P10 depends upon the event in which it occurs. | 10 | NUMBER |
| Parent_Op_ID[1] | Parent operation | 44 | NUMBER |
| PGA_Memory | Process Global Area memory | 101 | NUMBER |
| Physical Reads | Number of blocks read from disk | 105 | NUMBER |
| Position[1] | Position within events having same parent operation | 45 | NUMBER |
| Position_ID[2] | Position of the operation within the execution plan for a statement | 28 | NUMBER |

*Table B–4 Oracle Server Items (Cont.)*

| Item Name | Description | Item Number | Formatted Datatype |
|---|---|---|---|
| Redo_Entries | Number of redo entries made by process | 106 | NUMBER |
| Redo_Size | Size of redo entries | 107 | NUMBER |
| Row_Count | Number of rows processed | 29 | NUMBER |
| Schema_UID | Schema user ID | 37 | NUMBER |
| Session_Index | Oracle session ID | 12 | NUMBER |
| Session_Serial | Session serial number | 13 | NUMBER |
| SID | Text version of session ID | 17 | VARCHAR2(255) |
| Sort_Disk | Number of disk sorts performed | 110 | NUMBER |
| Sort_Memory | Number of memory sorts performed | 109 | NUMBER |
| Sort_Rows | Total number of rows sorted | 111 | NUMBER |
| SQL_Text | Text of SQL statement | 31 | LONG |
| SQL_Text_Hash | Pointer to SQL statement | 26 | NUMBER |
| SQL_Text_Segment | Address of SQL text | 30 | NUMBER |
| T_Scan_Rows_Got | Rows processed during full table scans | 108 | NUMBER |
| TX_ID | Unique identifier for a transaction that consists of rollback segment number, slot number, and wrap number | 41 | VARCHAR2(18) |
| TX_SO_Addr | The address of the transaction state object | 40 | VARCHAR2(16) |
| TX_Type | Type of the transaction. Value is a bitmap (for example, 2 active transaction, 0X10 space transaction, 0X20 recursive transaction). | 39 | NUMBER |
| UGA_Memory | User Global Area session memory | 100 | NUMBER |
| Wait_Time | Elapsed time, in hundredths of seconds, for the wait event | 21 | NUMBER |

[1] Item specific to Oracle Server release 8.0.2 and higher

[2] Replaced by Operation_ID for Oracle Server release 8.0.2 and higher

## Items Associated with Each Event

The following sections describe each event in more detail and provide tables that list the items associated with each event. For item descriptions, refer to Table B–4.

> **Note:** Prior to Oracle Server release 8.0.5, cross-product items 1-5 were set by the server code. Starting with Oracle Server release 8.0.5, cross-product item 6 was added.

When you format data, Oracle Trace creates a table for each event collected. The name of the event data table is **V_vendor#_F_product#_E_event#_version**, where **version** is the number of the Oracle Server release. Any periods in the product version are replaced with underscores. You can use the otrcsyn.sql script to create synonyms for these tables.

> **Note:** The following tables use Oracle7 Server names for example purposes.

The Oracle Trace formatter creates a column for each event item. For point events, the column name is the same as the item name. For duration events, the items for the start event have _START appended to the item name and the items for the end event have _END appended to the item name.

The formatter automatically includes additional columns for collection number, process identifier, and timestamp information as described in Table B–5.

***Table B–5 Additional Columns Included by Oracle Trace Formatter***

| Column Name | Description | Datatype |
|---|---|---|
| collection_ID | collection number, automatically assigned by the formatter | NUMBER(4) |
| epid | process ID | NUMBER(8) |
| timestamp | logged time for point events | DATE |
| timestamp_nano | fraction of seconds of logged time for point events | NUMBER |

*Table B–5 Additional Columns Included by Oracle Trace Formatter (Cont.)*

| Column Name | Description | Datatype |
|---|---|---|
| timestamp_start | duration event start time | DATE |
| timestamp_nano_start | fraction of seconds of duration event start time | NUMBER |
| timestamp_end | duration event end time | DATE |
| timestamp_nano_end | fraction of seconds of duration event end time | NUMBER |

## Event Statistics Block

Items relating to database performance appear in several events. For convenience, these items are referenced as the Event Statistics Block. The items in the Event Statistics block are shown in Table B–6:

*Table B–6 Event Statistics Block*

| | | |
|---|---|---|
| UGA_Memory | PGA_Memory | DB_Block_Change |
| DB_Block_Gets | Consistent_Gets | Physical_Reads |
| Redo_Entries | Redo_Size | T_Scan_Rows_Got |
| Sort_Memory | Sort_Disk | Sort_Rows |
| CPU_Session | Elapsed_Session | |

## Connection Event

The Connection event (event=1) records every time a connection is made to a database. Items associated with the Connection event are shown in Table B–7. The name of the formatter table is V_192216243_F_5_E_1_7_3 (for Oracle Server release 7.3).

*Table B–7 Items Associated with the Connection Event*

| | | |
|---|---|---|
| Session_Index | Session_Serial | Oracle_PID |
| Login_UID | Login_UName | SID |
| OS_UName | OS_Term | OS_Mach |
| OS_Image | Cross-Product Items 1-6 | |

The Oracle Server uses the combination of Session_Index and Session_Serial to uniquely identify a connection. Net8 uses the connection ID, stored in CROSS_FAC_3, to uniquely identify a connection.

## Disconnect Event

The Disconnect event records every time a database disconnection is made. Items associated with the Disconnect event are shown in Table B–8. The name of the formatter table is V_192216243_F_5_E_2_7_3.

***Table B–8 Items Associated with the Disconnect Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | Event Statistics Block |
| Oracle_PID | Cross-Product Items 1-6 | |

A Disconnect event will correspond to at most one Connection event. Therefore, the same fields uniquely identify a disconnect: either the combination of Session_Index and Session_Serial, or CROSS_FAC_3.

## ErrorStack Event

The ErrorStack event identifies the process that has the error. Items associated with the ErrorStack event are shown in Table B–9. The name of the formatter table is V_192216243_F_5_E_3_7_3.

***Table B–9 Items Associated with the ErrorStack Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | Oracle_PID |
| P1 | P2 | P3 |
| P4 | P5 | P6 |
| P7 | P8 | Cross-Product Items 1-6 |

The ErrorStack event does not have an explicit identifier. The combination of Session_Index, Session_Serial, Timestamp, and Timestamp_Nano should uniquely identify a specific ErrorStack event.

## Migration Event

The Migration event is logged each time a session migrates to a shared server process. The name of the formatter table is V_192216243_F_5_E_4_7_3. This event was disabled for Oracle Server release 7.3.2, but is enabled for all releases after 7.3.2.

Items associated with the Migration event are shown in Table B–10.

***Table B–10 Items Associated with the Migration Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | Oracle_PID |
| Cross-Product Items 1-6 | | |

The Migration event does not have an explicit identifier. The combination of Session_Index, Session_Serial, Timestamp, and Timestamp_Nano should uniquely identify a specific Migration event.

## ApplReg Event

The ApplReg event (event=5) registers with Oracle Trace where the application is at a certain point in time. Items associated with the ApplReg event are shown in Table B–11. The name of the formatter table is V_192216243_F_5_E_5_7_3.

***Table B–11 Items Associated with the ApplReg Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | App_Module |
| App_Action | Cross-Product Items 1-6 | |

The ApplReg event does not have an explicit identifier. The combination of Session_Index, Session_Serial, Timestamp, and Timestamp_Nano should uniquely identify a specific ApplReg event.

## RowSource Event

The RowSource event logs the number of rows processed by a single row source within an execution plan. Items associated with the RowSource event are shown in Table B–12. The name of the formatter table is V_192216243_F_5_E_6_7_3.

***Table B–12 Items Associated with the RowSource Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | Cursor_Number |
| Position_ID | Row_Count | Cross-Product Items 1-5 |

The combination of Session_Index, Session_Serial, Cursor_Number, and Position_ID uniquely identifies a RowSource event.

## RowSource Event (Specific to Oracle Server Release 8.0.2 and Higher)

The RowSource event logs the number of rows processed by a single row source within an execution plan. Items associated with the RowSource event for Oracle Server release 8.0.2 or higher are shown in Table B–13. The name of the formatter table is V_192216243_F_5_E_6_8_0.

*Table B–13 Items Associated with the RowSource Event*

| | | |
|---|---|---|
| Session_Index | Session_Serial | Cursor_Number |
| Operation_ID | Row_Count | Parent_Op_ID |
| Position | Object_ID | Operation |
| Cross-Product Items 1-6 | | |

The combination of Session_Index, Session_Serial, Cursor_Number, and Operation_ID uniquely identifies a RowSource event.

**Note:** The text in the Operation item is equivalent to information about the execution plan, which is similar to data that can be obtained by running explain plan.

## SQLSegment Event

The SQLSegment event is a description of a SQL statement. Items associated with the SQLSegment event are shown in Table B–14. The name of the formatter table is V_192216243_F_5_E_7_7_3.

*Table B–14 Items Associated with the SQLSegment Event*

| | | |
|---|---|---|
| Session_Index | Session_Serial | Cursor_Number |
| SQL_Text_Hash | Lib_Cache_Addr | SQL_Text_Segment |
| SQL_Text | Cross-Product Items 1-6 | |

A SQL segment does not have an explicit identifier. The SQL_Text_Hash field will always be the same for each occurrence of a SQL statement but multiple statements can have the same hash value. If a statement is forced out of the library cache and then swapped back in, the same statement can have multiple values for Lib_Cache_Addr. The combination of Session_Index, Session_Serial, SQL_Text_Hash, and Lib_Cache_Addr should usually identify a particular SQL statement for a session. If you add Cursor_Number, you will identify a particular occurrence of a SQL statement within the session.

## Wait Event

The Wait event shows the total waiting time in hundredths of seconds for all responses. Items associated with the Wait event are shown in Table B–15. The name of the formatter table is V_192216243_F_5_E_13_7_3.

***Table B–15 Items Associated with the Wait Event***

| | | |
|---|---|---|
| Session_Index | Session_Serial | Wait_Time |
| P1 | P2 | P3 |
| Description | Cross-Product Items 1-6 | |

The Wait event does not have an explicit identifier. The combination of Session_Index, Session_Serial, Description, Timestamp, and Timestamp_Nano should uniquely identify a specific Wait event.

## Parse Event

The Parse event records the start and end of the parsing phase during the processing of a SQL statement. The parsing phase occurs when the SQL text is read in and broken down (parsed) into its various components. Tables and fields are identified, as well as which fields are sort criteria and which information needs to be returned. Items associated with the Parse event are shown in Table B–16. The name of the formatter table is V_192216243_F_5_E_8_7_3.

***Table B–16 Items Associated with the Parse Event***

| | | |
|---|---|---|
| | **Items for Start of Parse Event** | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Resource Items | Cross-Product Items 1-6 |
| | **Items for End of Parse Event** | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Depth | Missed |

*Table B–16 Items Associated with the Parse Event (Cont.)*

| | | |
|---|---|---|
| Oracle_Cmd_Type | Optimizer_Mode | Current_UID |
| Schema_UID | SQL_Text_Hash | Lib_Cache_Addr |
| Resource Items | | |

The combination of Session_Index, Session_Serial, Cursor_Number, and SQL_Text_Hash uniquely identifies a specific Parse event.

## Execute Event

The Execute event is where the query plan is executed. That is, the parsed input is analyzed to determine exact access methods for retrieving the data, and the data is prepared for fetch if necessary. Items associated with the Execute event are shown in Table B–17. The name of the formatter table is V_192216243_F_5_E_9_7_3.

*Table B–17 Items Associated with the Execute Event*

| | | |
|---|---|---|
| | **Items for Start of Execute Event** | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Resource Items | Cross-Product Items 1-6 |
| | **Items for End of Execute Event** | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Depth | Missed |
| Row_Count | SQL_Text_Hash | Lib_Cache_Addr |
| Resource Items | | |

The combination of Session_Index, Session_Serial, Cursor_Number, and SQL_Text_Hash uniquely identifies a specific Execute event.

## Fetch Event

The Fetch event is the actual return of the data. Multiple fetches can be performed for the same statement to retrieve all the data. Items associated with the Fetch event are shown in Table B–18. The name of the formatter table is V_192216243_F_5_E_10_7_3.

*Table B–18 Items Associated with the Fetch Event*

| | | |
|---|---|---|
| **Items for Start of Fetch Event** | | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Resource Items | Cross-Product Items 1-6 |
| **Items for End of Fetch Event** | | |
| Session_Index | Session_Serial | Event Statistics Block |
| Cursor_Number | Depth | Row_Count |
| End_of_Fetch | SQL_Text_Hash | Lib_Cache_Addr |
| Resource Items | | |

The combination of Session_Index, Session_Serial, Cursor_Number, SQL_Text_Hash, Timestamp, and Timestamp_Nano uniquely identifies a specific Fetch event.

## LogicalTX Event

The LogicalTX event logs the start and end of a logical transaction (that is, statements issued that may cause a change to the database status). Items associated with the LogicalTX event are shown in Table B–19. The name of the formatter table is V_192216243_F_5_E_11_7_3.

*Table B–19 Items Associated with the LogicalTX Event*

| | | |
|---|---|---|
| **Items for Start of LogicalTX Event** | | |
| Session_Index | Session_Serial | Event Statistics Block |
| TX_Type | TX_SO_Addr | Resource Items |
| Cross-Product Items 1-6 | | |
| **Items for End of LogicalTX Event** | | |
| Session_Index | Session_Serial | Event Statistics Block |
| TX_Type | TX_SO_Addr | Resource Items |

The transaction identifier stored in CROSS_FAC_4 should uniquely identify a specific transaction. Or, use Session_Index, Session_Serial, and TX_SO_Addr.

## PhysicalTX Event

The PhysicalTX event logs the start and end of a physical transaction (that is, statements issued that caused a change in database status). Items associated with the PhysicalTX event are shown in Table B–20. The name of the formatter table is V_192216243_F_5_E_12_7_3.

*Table B–20 Items Associated with the PhysicalTX Event*

| Items for Start of PhysicalTX Event | | |
|---|---|---|
| Session_Index | Session_Serial | Event Statistics Block |
| TX_Type | TX_ID | Resource Items |
| Cross-Product Items 1-6 | | |
| **Items for End of PhysicalTX Event** | | |
| Session_Index | Session_Serial | Event Statistics Block |
| TX_Type | TX_ID | Commit_Abort |
| Resource Items | | |

The transaction identifier stored in CROSS_FAC_4 should uniquely identify a specific transaction.

# C

# Troubleshooting Oracle Trace

This appendix lists the problems you might encounter while using Oracle Trace. Solutions to these problems are provided, as well as a section on how to isolate problems encountered while using Oracle Trace. Sections in this appendix include:

- *Typical Problems Encountered While Using Oracle Trace*
- *Server Environment*
- *Oracle Enterprise Manager Configuration*
- *Network Configuration*
- *Oracle Intelligent Agent (UNIX Specific)*
- *Oracle Intelligent Agent (NT Specific)*
- *Database Configuration*
- *Oracle Trace Configuration*
- *Isolating an Oracle Trace Problem*

## Typical Problems Encountered While Using Oracle Trace

Table C–1 lists the symptoms Oracle Trace can exhibit, provides solutions, and refers you to the appropriate section for additional information. The table is also divided into sections which reflect how you would use Oracle Trace in the context of a collection.

***Table C–1 Typical Problems Encountered While Using Oracle Trace***

| Symptom | Solution |
|---|---|
| **Prior to Scheduling Collection** | |

*Table C–1 Typical Problems Encountered While Using Oracle Trace (Cont.)*

| Symptom | Solution |
|---|---|
| "Authentication User" error during product discovery. | Preferences for node are invalid or not set.<br>See "Oracle Enterprise Manager Configuration" on page C-6. |
| Products not discovered. | Product definition files might be missing.<br>See "Oracle Trace Configuration" on page C-12. |
| "Failed to connect to agent" during product discovery. | Cannot connect to Oracle Intelligent Agent.<br>See "Network Configuration" on page C-8.<br>See "Oracle Intelligent Agent (UNIX Specific)" on page C-8.<br>See "Oracle Intelligent Agent (NT Specific)" on page C-9. |
| Database missing from product discovery. | Verify oratab file is on the server. If file exists, verify contents of oratab file.<br>See "Oracle Intelligent Agent (UNIX Specific)" on page C-8. |
| Oracle Trace discovery fails while processing files for a 7.3 ORACLE_HOME. | Oracle Trace discovery process may have encountered a pre-7.3.3 otrccol image which is not compatible with Oracle Trace Manager.<br>See "Server Environment" on page C-5. |
| **Creating Collection** | |
| Cannot connect to Show User List. | Edit the tnsnames.ora file and add a service name that matches the service name known to the Oracle Enterprise Manager console.<br>See "Failure Connecting to Show User List" on page C-16. |
| **Scheduling Collection** | |
| Cannot start Oracle Trace from Oracle Enterprise Manager console. | Start Oracle Trace using Start menu or reinstall Oracle Trace. |
| No nodes listed in Oracle Trace Manager. | In Oracle Enterprise Manager console, discover nodes that you want to manage by choosing Navigator=> Discover Nodes, then perform product discovery in Oracle Trace Manager.<br>See "Oracle Enterprise Manager Configuration" on page C-6. |

*Table C–1 Typical Problems Encountered While Using Oracle Trace (Cont.)*

| Symptom | Solution |
|---|---|
| "Authentication User" error during collection creation. | Preferences for node are invalid or not set.<br>See "Oracle Enterprise Manager Configuration" on page C-6. |
| "Could not read file <filespec>". | File not in expected location.<br>See "Oracle Trace Configuration" on page C-12. |
| Error starting/stopping Oracle7 database collection. | There are missing database stored procedures that Oracle Trace uses to start and stop an Oracle7 database collection.<br>See "Stored Procedures" on page C-22. |
| Oracle Trace Manager shows collection with status ’Created’ and Oracle Enterprise Manager Job system shows collection "awaiting agent confirmation". | Cannot connect to Oracle Intelligent Agent.<br>See "Network Configuration" on page C-8.<br>See "Oracle Intelligent Agent (UNIX Specific)" on page C-8.<br>See "Oracle Intelligent Agent (NT Specific)" on page C-9. |
| Oracle Trace Manager shows collection with status ’Created’ and Oracle Enterprise Manager Job system shows collection with changing status. | Oracle Intelligent Agent messages are not returned to Oracle Trace Manager.<br>Exit and restart Oracle Trace Manager.<br>Exit and restart Oracle Enterprise Manager console and Oracle Trace Manager. |
| **Executing Collection** | |
| Error finding executable or command. | Preference setting node has invalid path.<br>See "Server Environment" on page C-5. |
| "Could not read file <filespec>". | File not in expected location.<br>See "Oracle Trace Configuration" on page C-12. |
| Network timeout. | See "Network Configuration" on page C-8. |
| Error accessing memory for collections running on NT server nodes. | See "Oracle Trace Could Not Access Memory" on page C-17. |
| Error starting collection on server node. | See "Database Configuration" on page C-9.<br>See "Oracle Trace Configuration" on page C-12. |

*Table C–1 Typical Problems Encountered While Using Oracle Trace (Cont.)*

| Symptom | Solution |
|---|---|
| Error starting NT database collection, with memory mapping error. | See "Oracle Trace Configuration" on page C-12 for discussion of mismatch between NT database and Oracle Trace Collection Services. |
| Collection does not end at scheduled time. | Possibly out of disk space or Oracle Intelligent Agent is not available.<br>See "Server Environment" on page C-5.<br>See "Network Configuration" on page C-8.<br>See "Oracle Intelligent Agent (UNIX Specific)" on page C-8.<br>See "Oracle Intelligent Agent (NT Specific)" on page C-9. |
| **Formatting Collection** | |
| Format gets "unable to resolve service name" error. | Database not available or invalid tnsnames.ora file on server.<br>See "Network Configuration" on page C-8.<br>See "Database Configuration" on page C-9. |
| Format gets "invalid username/password" error. | Preferences setting for database is invalid or not set. |
| Format gets "table does not exist" error (Oracle Server release 7.3.x and earlier). | Formatter tables not created.<br>See "Database Configuration" on page C-9.<br>See "Formatter Tables" on page C-18. |
| Format gets error extending tablespace. | Insufficient space in target tablespace in the database to which you are formatting. Allocate additional space to tablespace and reformat.<br>See "Collection Is Too Large" on page C-14. |
| Format gets insufficient privilege error. | See "Could Not View Oracle Trace Formatted Data" on page C-13. |
| **Examining Collection Data** | |
| Collection contains no data. | Oracle Trace interface to database is incorrectly configured or there are too many collections running in parallel.<br>See "Database Configuration" on page C-9.<br>See "Collection Is Empty" on page C-16. |

*Table C–1 Typical Problems Encountered While Using Oracle Trace (Cont.)*

| Symptom | Solution |
|---|---|
| SQL statement not captured in collection. | See "Missing SQL Statement from Collection" on page C-14. |
| Wait times were not collected (for ALL event set). | See "Wait Times Were Not Collected" on page C-14. |
| Collection has additional data but Data Viewer summaries are unchanged. | See "Additional Formatted Data Is Missing from Data Viewer" on page C-14. |
| Data Viewer gets errors computing statistics. | See "Need to Restore Formatted Data" on page C-14. |
| Data Viewer performance is poor. | See "Poor Performance" on page C-15. |

## Server Environment

If you suspect a server environment problem, verify:

- That the account used by the Oracle Enterprise Manager Job system on the server node has:
  - The ORACLE_HOME and the ORACLE_SID (Oracle System Identifier) environment variables defined.
  - The $ORACLE_HOME/bin directory is included in the path for the account.
  - Write access to the $ORACLE_HOME/otrace/admin/cdf directory
- That the server node has sufficient disk space for the collection output files. If there is not sufficient disk space, the collection stops. Use the Oracle Trace Collection wizard to restrict the maximum file size. Because collection files can get large, collect small amounts of data.

  To solve the immediate problem, stop the collection, and free up space so Oracle Trace can end the collection.

  Initially limiting the collection to specific users and/or Wait events will also help to limit the amount of data collected. Limiting users and Wait Events is available for Oracle Server releases 8.0.4 and higher.

- That your session does not participate in more than five collections. Sessions log data for the five most recent collections. Thus, if you have more than five collections, data will be missing for the oldest collection.
- That the otrccol image is compatible with Oracle Trace Manager. If a 7.3 $ORACLE_HOME/bin directory contains an otrccol image, determine the version of the image by executing the following command:

  ```
  $ORACLE_HOME/bin/otrccol version
  ```

  This command should return a small integer, for example 1 (for Oracle Server release 7.3.3) or 3 (for Oracle Server release 7.3.4); any error returned by the ottrcol version command indicates an otrccol image that is not compatible with Oracle Trace Manager. You can leave the otrccol image in the $ORACLE_ HOME/bin directory, but you must rename the image (for example, to otrccol732). If you rename the image, Oracle Trace will not attempt to use the image during the discovery process.

## Oracle Enterprise Manager Configuration

Problems covered in this section are also known as client environment problems. Some of the symptoms of a configuration problem are: nodes and databases have not been discovered or refreshed, user preferences appear to be invalid, and jobs are not running in the Oracle Enterprise Manager Job system.

Try the following:

- Discover nodes and databases by choosing Navigator=>Discover Nodes... from the Oracle Enterprise Manager console.
- Verify that user preferred credentials are correct. Set preferred credentials by choosing System=>Preferences... in the console. Then select the Preferred Credentials tab to set node and database access credentials.

  **Note:** Preferred credentials need to be set separately for each domain user. Check the preferences set for the domain user that you currently used to log into Oracle Trace.

  Verify that the user account is valid by either invoking SQL worksheet to test the database user account or rlogin to the node to test the node account.

  By invoking SQL worksheet using the username/password@service for the account, you are verifying the existence and accessibility of a database user.

By using rlogin to log into the node, you are verifying the node criteria. If the node is not available, check with your system support personnel regarding the availability of the account.

- Check the Oracle Enterprise Manager Job system especially if you are having problems running jobs. To create a test job, choose Job=>Create Job... from the console. Schedule a job to run an OS command (’ls’ on UNIX and ’dir’ on NT).

  In the Oracle Enterprise Manager Job System on NT, you can:

  1. Click the General tab. The Destination Type equals Node. Select the node on which the Oracle Intelligent Agent is located.
  2. Click the Tasks tab. From Available Tasks, choose Run OS Command from the Available Destinations list.
  3. Click the Parameters tab. On the Task Parameters section, type ’dir’ in the Command Edit control.
  4. Click the Submit button.

  If the job does not work, there is a problem with the Oracle Enterprise Manager Job system. Consult the Oracle Enterprise Manager console documentation for additional information.

- Verify that the Oracle Trace repository tables have been successfully created. You can verify the existence of the Oracle Trace repository tables using Oracle Server Manager or Oracle SQL*Plus Worksheet.

```
connect as repository <userid>
select table_name
from user_tables
where table_name like ’EPC_CLI%’;
```

- If you received the "Authentication User" message, you need to set the Oracle Enterprise Manager preferred credentials for the node to a valid user name and password with the appropriate privileges.

> **Note:** Preferred credentials must be set individually for each domain user.

If the account is on an NT node, the account must have the Advanced User Right ’Log on as a batch job’ enabled.

See the instructions on setting preferences in the *Oracle Enterprise Manager Configuration Guide*.

## Network Configuration

Some of the symptoms of a network configuration problem are: the network times out, error messages stating the system could not be found or the service name could not be resolved.

Try the following:

- If available on your system, run the Ping utility to verify that the underlying network communication exists. Ping the server from the client and ping the client from the server.
- Verify that the entries in the tnsnames.ora file on the client match the entries in the tnsnames. ora file on the server.

## Oracle Intelligent Agent (UNIX Specific)

Symptoms of an agent configuration problem are error messages stating the agent is not working or that the agent is not found. This section offers a few suggestions. Refer to the *Oracle Enterprise Manager Configuration Guide* for a more in-depth explanation of the Oracle Intelligent Agent.

Verify:

- Write access to the $ORACLE_HOME/network/agent directory or $ORACLE_HOME/net80/agent directory.
- That the $ORACLE_HOME/bin/dbsnmp agent executable is owned by the root account (applicable to Oracle Server release 8.0.4 and higher).
- That the $ORACLE_HOME/bin/dbsnmp agent executable has the ’sticky bit’ set as represented by the letter s in the following code example (applicable to Oracle Server release 8.0.4 and higher). Though the executable is owned by the root account, the ’sticky bit’ executes dbsnmp with the privileges of the invoking user, rather than with root privileges.

  ```
  ls -l dbsnmp
  -rwsr-s--x
  ```

  If the sticky bit is not set, run the following script as the root account:

  ```
  $ORACLE_HOME/orainst/root.sh
  ```

- The processes:
  - check Oracle Intelligent Agent status

    ```
    lsnrctl dbsnmp_status
    ```

  - Check for two dbsnmp processes

    ```
    % /usr/ucb/ps auxw | grep dbsnmp
    ```

- The contents of the oratab file (located in the /var/opt/oracle directory) and the listener.ora file. The oratab file is a UNIX-specific file created during installation and lists the installed or known databases on that node. Make sure that the oratab file does not contain entries for databases that no longer exist. The listener.ora file contains the names and locations of the databases you want to monitor. The *Oracle Enterprise Manager Configuration Guide* describes the proper settings for these files.
- The information in the services.ora file compares with the information in the listener.ora file. For example, if you add an item to the list in the listerner.ora file and did not restart the agent, the new item will not be found in the services.ora file. The services.ora file was created when the agent was started for the first time.
- If the agent will not start, refer to the *Oracle Enterprise Manager Configuration Guide* for possible solutions.

## Oracle Intelligent Agent (NT Specific)

Symptoms of an agent configuration problem are error messages stating the agent is not working or that the agent is not found.

Verify that the agent is up and running. Select Start=>Settings=>Control Panel from the Task bar. Double-click on Services. Start the agent if it is not started.

Also, examine the NT Event log for individual applications. This log may provide additional information to help solve the problem.

## Database Configuration

The symptom is that you scheduled a collection but no data is in the collection or there was an error formatting the data. Do the following.

- Verify that you can attach to the database, for example, from Oracle Server Manager or Oracle SQL*Plus Worksheet using the connect command:

  ```
  connect <username>/<password>@<service name>
  ```

  For earlier database versions on NT, the command for invoking the Oracle Server Manager is either *svrmgr30* or *svrmgr23*. On UNIX the command is *svrmgrl*.

- Verify that the database and listener are running on the server.

  ```
  test-sun% /usr/ucb/ps auxw | grep oracle
  oracle   10684  0.7  5.92736814784 ?       S 12:37:46  1:00 ora_ckpt_v805
  oracle   10678  0.0  6.42843215952 ?       S 12:37:46  0:00 ora_pmon_v805
  oracle   10680  0.0  6.02744014960 ?       S 12:37:46  0:00 ora_dbw0_v805
  oracle   10682  0.0  5.92739214848 ?       S 12:37:46  0:00 ora_lgwr_v805
  oracle   10686  0.0  6.42630416048 ?       S 12:37:46  0:00 ora_smon_v805
  oracle   10688  0.0  6.32628815920 ?       S 12:37:46  0:00 ora_reco_v805
  oracle   10809  0.0  3.211064 8112 ?       S 14:36:10  0:00
  /oracle/app/oracle/product/805/bin/tnslsnr LISTENER -inherit
  ```

- Verify that all required patches have been applied to the database and that the database has been upgraded.

- Verify init.ora parameters for Oracle Trace.

  - ORACLE_TRACE_ENABLE=TRUE for Oracle Server release 8.0 and higher.

  - If you schedule the collection through the command-line interface, supply the collection path (ORACLE_TRACE_FACILITY_PATH) and the collection name (ORACLE_TRACE_FACILITY_NAME). Make sure these parameter values are valid names and that the account running the collection has access to the specified directory and file.

  - If you change any init.ora parameters, the database must be restarted for the parameters to take effect.

- Check for stored procedures (for Oracle Server releases 7.3.x)

  To check for stored procedures using the Oracle Enterprise Manager console, use the Navigator and the following path:

  ```
  Networks=>Databases=><your database>=>Schema Objects=>Packages=>SYS
  ```

  Look for stored procedures starting with DBMS_ORACLE_TRACE_xxx.

To check for stored procedures using Oracle Server Manager or Oracle SQL*Plus Worksheet:

```
select object_name from dba_objects where object_name like '%TRACE%'
and object_type = 'PACKAGE';
OBJECT_NAME
DBMS_ORACLE_TRACE_AGENT
DBMS_ORACLE_TRACE_USER
2 rows selected.
```

- Check privileges of the database account. Using the Navigator in the Oracle Enterprise Manager console, choose:

  ```
  Networks=>Databases=><your database>=>Users=><dba account>=>System
  privileges
  ```

  See "Could Not View Oracle Trace Formatted Data" on page C-13 for the list of required privileges.

- Check for formatter tables (for Oracle Server releases 7.3.x)

  To check for formatter tables using the Oracle Enterprise Manager console, use the Navigator and the following path:

  ```
  Networks=>Databases=><your database>=>Users=><user id>=>Schema
  Objects=>tables
  ```

  Look for EPC_COLLECTION.

  To check for formatter tables using SQL Worksheet:

  ```
  connect <username>/<password>@<service name>
  describe epc_collection
  ```

  To create formatter tables, execute the otrcfmtc.sql script from the ORACLE_HOME of the Oracle Server that you connected to. You only need to run this script on Oracle Server release 7.x collections.

  ```
  @ORACLE_HOME/otrace/admin/otrcfmtc.sql
  ```

- Check tablespace usage and extents available using Oracle Storage Manager. There may not be enough space to hold the collection data.

- Verify that users do not have SYSTEM as default tablespace. Using the Navigator in Oracle Enterprise Manager:

  ```
  Networks=>Databases=><your database>=>Users
  ```

Right mouse click on the user ID you want to verify and select Quick Edit. The Tablespaces section of the General tab lists the default tablespace. The default tablespace should be a user tablespace.

## Oracle Trace Configuration

If you suspect an Oracle Trace configuration problem:

- Examine the EPC_ERROR.LOG file for details of any logged Oracle Trace errors.
- Look for the administration files on $ORACLE_HOME/otrace/admin (*.dat files) and run otrccref to recreate the Oracle Trace *.dat files if the files do not exist.
- Verify that the .fdf files are in the $ORACLE_HOME/otrace/admin/fdf directory.
- Verify that the TCL scripts are in the correct directory. The server files are located under $ORACLE_HOME/network/agent/jobs/oracle/otrace/general

  ```
  % cd $ORACLE_HOME/network/agent
  % find . -name "otrc*.tcl" -print
  ```

  The client file (otrcjob.tcl) is located in $ORACLE_HOME\Sysman\Scripts\Tcl directory.
- Verify the correct version of Oracle Trace Collection Services match the appropriate Oracle Server version

  ```
  % $ORACLE_HOME/bin/otrccol version
  ```

***Table C–2 Matching Releases of Oracle Trace Collection Services with Releases of Oracle Server***

| If the Returned Value Is: | Then the Command-Line Interface Release Is: |
|---|---|
| 1 | 733 |
| 2 | 803 |
| 3 | 734 |
| 4 | 804 |
| 5 | 805 |

***Table C–2 Matching Releases of Oracle Trace Collection Services with Releases of Oracle Server (Cont.)***

| If the Returned Value Is: | Then the Command-Line Interface Release Is: |
|---|---|
| 6 | 813 |
| 7 | 814 |
| 8 | 815 |

- To verify that a collection is currently running, use the command-line interface to check the status:

  ```
  % $ORACLE_HOME/bin/otrccol check <collection_name>
  ```

- To verify the Oracle Trace configuration, use the command-line interface to start, stop, and format collections.

  ```
  otrccol start <parameters>
  otrccol stop <parameters>
  otrccol format <parameters>
  ```

  Refer to the *Oracle Enterprise Manager Oracle Trace Developer's Guide* for all the possible parameters.

## Data Viewer Problems

The following section deals with problems specific to Data Viewer.

### Could Not View Oracle Trace Formatted Data

To view Oracle Trace formatted data using the Data Viewer, a user account needs the following system privileges:

CREATE SESSION
CREATE SEQUENCE
CREATE TABLE
CREATE PROCEDURE
CREATE ANY INDEX
EXECUTE ANY PROCEDURE
DELETE ANY TABLE
DROP ANY INDEX
DROP ANY SEQUENCE

UNLIMITED TABLESPACE

### Need to Restore Formatted Data

Data Viewer adds statistical information to your formatted data. To restore your formatted data to the exact state it was in prior to running Data Viewer, run the ORACLE_HOME\Sysman\Admin\TdvDrop.sql script. Reasons for restoring data include:

- You encountered an insufficient privilege error prior to or during computing statistics.
- You received a database error creating a tablespace extent.

  ```
  ORA-01658: unable to create INITIAL extent for segment in tablespace
  MY_TABLESPACE
  ```

- You encountered an unexpected error that appears to be non-recoverable.

### Wait Times Were Not Collected

Wait times are collected only if the INIT<sid>.ORA parameter, TIMED_ STATISTICS, is set to TRUE. When you change this database parameter, you must stop and restart the database for the change to take effect.

### Additional Formatted Data Is Missing from Data Viewer

The first time Data Viewer views a collection, you can compute SQL statistics for all queries currently in the collection. Data Viewer does not detect new data added to a collection by a partial format while Data Viewer is viewing that collection. If you choose to partially format your collected data, you can force Data Viewer to recompute statistics using both existing and newly collected data by choosing Options=>Recompute Statistics.

### Missing SQL Statement from Collection

If an expected SQL statement does not appear to be in your collection, it may be because the Oracle Trace auto-format function does not flush the collection buffer until the next time the database is accessed after the collection was stopped. This may result in a small amount of event data not reaching the collection file.

### Collection Is Too Large

There may be times when a collection is too large. Starting with Oracle Trace release 1.5.5 and Oracle Server release 8.0.4, you can collect data for specific users and specific Wait event types to minimize the size of the collection. Because the server is almost always waiting for a latch, lock, or resource, Wait event data for a brief collection can be quite extensive.

The Oracle Trace Collection Wizard walks you through limiting the collected data by individual users.

### Poor Performance

If multiple collections exist in one set of user tables, computing statistics and subsequent selects may take longer than necessary. Placing a large collection in a new database user account will improve performance.

### Table or View Does Not Exist (or No Data in Collection)

When a record is not written to the EPC_FACILITY_REGISTRATION table, the user may see an error like the following:

```
XP-21016: A database error has occurred:
SELECT DISTINCT FACILITY_NUMBER, FACILITY_VERSION, VENDOR
FROM EPC_FACILITY_REGISTRATION WHERE COLLECTION_ID - :1
ORA-00942: table or view does not exist
```

Removing the ’Filtering by User’ option on collections targeting Oracle Server release 8.0.4 databases will correct this problem for future collections.

Manually adding an EPC_FACILITY_REGISTRATION record for the collection allows Trace Data Viewer to see that collection’s formatted data. For example, insert EPC_FACILITY_REGISTRATION record with the following values:

```
Collection_ID:    123       [Look in formatted data’s EPC_COLLECTION table
                             for the collection_id column value that matches
                             your collection_name]
Vendor:           192216243 [Hard code this value]
Facility_number:  5         [Hard code this value]
Facility_version: ’8.0’     [or ’7.3’ for a collection against a 7.3 database
                             or ’8.1’ for a collection against an 8.1 database]
```

## Oracle Trace Manager Problems

The following section deals with problems specific to Oracle Trace Manager.

### Failure Connecting to Show User List

If the user name and password you use to connect to the target collection database are correct, the problem may be that the service name used for connection is not the same name as your service name for the target collection database.

The service name is set to the service name as it is known to the Oracle Enterprise Manager console. The console gets the service name of discovered databases from the agent.

If the service name does not match the service name in your tnsnames.ora file, you will not be able to connect to the database when you click the Show User List button.

Edit your tnsnames.ora file and add a service name that matches the service name known to the Oracle Enterprise Manager console.

### Format of Collection Failed

There are a number of reasons why this can occur.

- Ran out of space in the database; tablespace needs extending.
- Invalid user name/password.
- The otrcfmtc.sql script has not been run to create the formatter tables. The script is located in $ORACLE_HOME\otrace*nn*\admin. (This step was required for releases prior to Oracle Trace Collection Services releases 7.3.4 and 8.0.4.) For Oracle Trace releases 1.3.6 and higher, run the vobsh command to create the formatter tables. Refer to the *Oracle Enterprise Manager Configuration Guide* for additional information about the vobsh command.
- If the formatter tables are not local to the node running the Oracle Enterprise Manager console, check that the node where the collection was performed does have a TNS names entry for the format database.
- If the collection has already been formatted, it will not be formatted again. To format the collection, disable the Partial Format option.
- Preferred credentials must use a service name from the list of Oracle Enterprise Manager service names. That service name must be in the tnsnames.ora file for the server the data was collected on.

### Collection Is Empty

In Oracle8 databases, the ORACLE_TRACE_ENABLE parameter in the INIT*sid*.ORA file on the server must be set to TRUE before the database is started.

(In previous versions of Oracle Server, the ORACLE_TRACE_ENABLE parameter was set to FALSE.) You can also see this problem if there are too many collections running concurrently.

Refer to "Database Configuration" on page C-9 for additional information.

### Oracle Trace Could Not Access Memory

On Windows NT systems, if you are running Oracle Trace collections and an error occurs indicating Oracle Trace could not access memory, the collect.dat file has become full. You must create a new .dat file by running the otrccref.exe image located in the $ORACLE_HOME/bin directory. However, database services must be shutdown to release the collect.dat file for the otrccref script to be able to create the new collect.dat file. You can also increase the number of collect.dat records above the default of 36 records (for example, otrccref -c50).

# Isolating an Oracle Trace Problem

A good approach in isolating a problem is to perform tests on each component of Oracle Trace. The components are:

- Oracle Trace Manager (graphical user interface)
  - Oracle Trace
  - Oracle Intelligent Agent and Oracle Enterprise Manager Job system
- Oracle Trace Collection Services
  - init.ora parameters
  - Stored procedures (for Oracle Server release 7.3.x only)
  - Oracle Trace Command-Line Interface (CLI)

There are two Oracle Trace mechanisms that are valuable when troubleshooting: Oracle Trace logging window (see "Oracle Trace Manager Logging" on page C-23) and EPC_ERROR.LOG file (see "EPC_ERROR.LOG File" on page C-23).

## Isolating a Problem in Oracle Trace Manager

You can test Oracle Trace Manager to determine if the problem is within the graphical user interface. Turn on Oracle Trace logging to see the communication within Oracle Trace Manager.

Oracle Trace Manager performs the following:

1. Takes input from the user.
2. Packages the information from the user into a temporary input parameters file. Oracle Trace Manager uses the Oracle Enterprise Manager Job system to schedule specified tasks. Within the Oracle Intelligent Agent, the otrcjob.tcl script then runs the job to be performed on the database server.

   The otrcjob.tcl script is a general purpose tcl script. It manages the other tcl scripts by taking input parameters plus those supplied by the temporary parameter files and running one of the other task specific tcl scripts.

   Each of the other tcl scripts is used to send the information required by the CLI images to perform the specific task against the database: start, stop, delete, or format a collection.
3. The jobs submitted run the CLI images to start and stop collections, format the collection data, and delete the collection.
4. The jobs manage the return of the information from the CLI to Oracle Trace Manager.

To check and see if the communications is set up correctly, you can run the Oracle Enterprise Manager Job system independently. You may also want to ping the server from the client and ping the client from the server.

### Oracle Enterprise Manager Job System

You can use the Oracle Enterprise Manager Job system to test if the communication between Oracle Management Server and the Oracle Intelligent Agent is working. For step by step instructions, see "Oracle Enterprise Manager Configuration" on page C-6.

### Formatter Tables

If you format a collection, Oracle Trace will format the data into your repository database and schema by default. The format operation requires repository tables to store collection information.

You can check that the formatter tables have been created.

You can query the database and look for the tables “EPC_COLLECTION”. The tables that start with “EPC_CLI” are the repository tables. All the other tables that start with “EPC_” are the formatter tables.

An easy way to find the formatter tables is to use Oracle Enterprise Manager Schema Manager to graphically see if the formatter tables are present.

Oracle Server releases 7.3.4 and 8.0.4 and later automatically create the formatter tables. Prior to Oracle Server releases 7.3.4 and 8.0.4, you must run the otrcfmtc.sql script from Oracle Server Manager or Oracle SQL*Plus Worksheet as the user who will be formatting the data. To ensure version compatibility, use the otrcfmtc.sql script located in the ORACLE_HOME of the destination server.

The otrcfmtc.sql script is located in the $ORACLE_HOME/otrace*nn*/admin directory (where *nn* represents the version of Oracle Trace).

Formatting error might be due to one of the following causes:

1. The user did not run the script to create the formatter tables (valid for releases of Oracle Server prior to 7.3.4 and 8.0.4).
2. The formatter tables were not created by the same user ID that was used when the collection was created (valid for releases of Oracle Server prior to 7.3.4 and 8.0.4).
3. There is a version mismatch. If the user formats to a given schema with an Oracle Server release 8.0.4 formatter, then later tries to format to the same schema using an older formatter, for example Oracle Server release 7.3.x, an error occurs. For additional information see "New otrcfmt Image for Oracle Server Releases 7.3.4 and 8.0.4" on page A-8.

## Isolating a Problem in Oracle Trace Collection Services

Here are a few suggestions to help you isolate problems in the Oracle Trace Collection Services.

### Oracle Trace Command-Line Interface

Oracle Intelligent Agent uses the command-line interface to access the Oracle Trace Collection Services. You can run the CLI to determine if the Oracle Trace Collection Services are functioning properly.

The CLI is located on the same server as the database that you are running the collection against. If you have multiple ORACLE_HOMEs, you could have multiple CLIs, that is, one CLI per ORACLE_HOME. Verify that your path is pointing to the correct CLI.

To test the CLI:

1. Run the CLI directly on the server node to test if the collection services are working properly.

   a. CLI needs to run from a privileged account, for example, the Oracle operating system user account.

   b. The Oracle Home and SID environmental variables must be set properly.

      To check settings on UNIX:

      ```
      printenv ORACLE_HOME
      printenv ORACLE_SID
      ```

      To set settings on UNIX:

      ```
      setenv ORACLE_HOME <path>
      setenv ORACLE_SID <sid>
      ```

      There should be one CLI per ORACLE_HOME. For example, if you have two Oracle Server release 7.3.3 instances sharing the same ORACLE_HOME, there should be only one CLI.

2. Verify that the CLI release matches the database server release.

   The Oracle Trace Collection Services files may be overwritten on NT by other product installations. If this occurs, your Oracle Trace Collection Services files will no longer match the version required by your Oracle Server database and you may receive a memory mapping error.

   On the server node, type the following command on the operating system command line:

   ```
   $ORACLE_HOME/bin/otrccol version
   ```

   See Table C–2 for the possible results. If the versions do not match, call Oracle Support for additional assistance.

3. Verify if the collection name has not already been used before you start the collection. Look in the .cdf and .dat directories $ORACLE_HOME\otrace*nn*\admin on NT ($ORACLE_HOME/otrace/admin on UNIX).

4. Prepare your input file as follows:

   ```
   <collection_name>.inp

   col_name= col_name (no spaces before equal sign and 1 space after equal
   sign)
   dat_file= col_name.dat
   cdf_file= col_name.cdf
   fdf_file= oraclee.fdf (whatever .fdf file you wish to use)
   regid=1 192216243 0 0 5 <your_service_name> [for Oracle8 use <sid>]
   ```

```
resubmit= 0 (1 only if resubmission)
```

5. If you want to generate database activity for this collection, connect to the database.
   - For Oracle Server release 7.3.x, connect to the service before you create your collection.
   - For Oracle Server release 8.0, you can connect to the database anytime and the processes will be registered.
6. To test the entire CLI process, start a collection, verify the collection is running, and stop the collection. For example:

   a. To start a collection:

   ```
   otrccol start <job_number> <collection_name.inp> (see step 4
   where you can make up your own input parameter file)
   ```

   You should see the response “collection started”.

   A .dat file and a .cdf file will be created in one of the following directories: $ORACLE_HOME/otrace/admin/cdf or the directory specified by the ORACLE_TRACE_COLLECTION_PATH parameter in your init<sid>.ora file.

   b. Verify that the collection was created.

   ```
   otrccol check <collection_name>
   ```

   The collection should show as active.

   c. To stop a collection:

   ```
   otrccol stop <job_number> <collection_name.inp>
   ```

   After you run the stop command your collection will not show as active.

   If there are any errors, the EPC_ERROR.LOG file will be written to the same directory from which you run the otrccol command.

7. Test that the format works properly. Choose a schema in a database with enough room to hold the data.

   a. Verify that the formatter tables have been created by querying the schema to see if there are tables prefixed with ‘EPC_’. The Oracle Trace repository tables are prefixed with ‘EPC_CLI’ and the formatter tables are prefixed with ’EPC_’. You can also use Oracle Schema Manager to look at the tables.

**Note:** For Oracle Trace Collection Services releases prior to 7.3.4 and 8.0.4, you need to manually run the otrcfmtc.sql script located on $ORACLE_HOME/otrace/admin to create the formatter tables.

**b.** Attempt to format a collection using the command-line interface.

```
otrccol format <format_input_file>
```

The format_input_file contains the following input parameters:

```
username= <db username>
password= <db pw>
service= <db service name>
cdf_file= <collection_name>.cdf
full_format= <0/1>
```

Full format value of 1 formats all the data in the data file, 0 formats only the data that has not previously been formatted.

### Stored Procedures

If the attempt to collect Oracle Trace data for an Oracle7 database results in the message "Error starting/stopping Oracle7 database collection," this may be due to missing database stored procedures that Oracle Trace uses to start and stop Oracle7 collections.

To verify that the stored procedures are present:

- Select the database in the Navigator tree in the Oracle Enterprise Manager console.
- Select Schema Objects=>Packages=>SYS.

  Verify that DBMS_ORACLE_TRACE_USER and DBMS_ORACLE_TRACE_AGENT are listed. This is an issue only if you are using Oracle Server release 7.3.x.

Prior to Oracle Server release 8.0.3, Oracle Trace required that stored procedures be installed on the database. These SQL scripts may be automatically run during database installation depending on the platform-specific installation procedures. If they are not executed during database installation, you must run these scripts manually. You can add these stored procedures to the database by running the otrcsvr.sql script from $ORACLE_HOME\otrace*nn*\admin on NT ($ORACLE_HOME/otrace/admin on UNIX) from a privileged database account (SYS or

INTERNAL). To run the script, set the default to the path were the script is located. This script runs other scripts that do not have the path specified. These other scripts fail if you are not in the directory where these scripts will run.

## Oracle Trace Manager Logging

There are two methods by which to capture activity taking place in the Oracle Trace Manager: logging window (/l option) and log file (/o option). Run Oracle Trace Manager with the /l option, the /o option, or both options activated. To activate the options, do the following:

1. Run the Oracle Enterprise Manager console.
2. From the Start menu, choose Run.
3. Assuming your ORACLE_HOME is C:\ORANT, in the Open field, type:

   ```
   C:\ORANT\BIN\EPC.EXE /l /o=<filename> [Do not leave any spaces
   between the equal sign and the filename.]
   ```

4. Click OK.

This displays the Oracle Trace Log window, which displays in-depth information about Oracle Trace Manager client/server processing. This information includes:

- Node and discovery information
- Information about jobs being submitted
- Detailed error messages

With the /o option activated, you automatically have a file copy of the information displayed in the logging window.

## EPC_ERROR.LOG File

The EPC_ERROR.LOG file provides information about the collection processing, specifically the Oracle Trace Collection Services errors.

The EPC_ERROR.LOG file is created in the current default directory of the Oracle Intelligent Agent when it runs the Oracle Trace Collection Services otrccol image.

Depending if you are running Oracle Trace from the Oracle Trace Manager or the command-line interface, the EPC_ERROR.LOG file can be located in one of the following server locations:

- $ORACLE_HOME or $ORACLE_HOME/network/agent (on UNIX)

- %ORACLE_HOME%\network\agent or %ORACLE_HOME%\net80\agent (on NT)
- $ORACLE_HOME\rdbms*nn* on NT ($ORACLE_HOME\rdbms on UNIX)
- In current working directory, if you are using the command-line interface

To find the file on UNIX, do the following:

```
% cd $ORACLE_HOME
% find . -name EPC_ERROR.LOG -print
```

> **Note:** On UNIX the EPC_ERROR.LOG file name is case sensitive and is in uppercase.

For general information about causes and actions for most Oracle Trace messages, see the *Oracle Enterprise Manager Messages Manual*.

## Information Needed When Reporting a Problem

If you have tried all the previously described suggestions and Oracle Trace is still not working, please call your local Oracle World Wide Support Center. When reporting a problem, have the following information available:

- Version of Oracle Trace
- Version of Oracle Intelligent Agent
- Operating system and version you are using for the server and the version of the Oracle Server
- Operating system you are using for the client and the version of the client
- Verify whether you have multiple ORACLE_HOME installations. If so, what Oracle components and versions are in each ORACLE_HOME installation?
- Steps to follow to reproduce the problem
- Log files (information from Oracle Trace Manager logging window and EPC_ERROR.LOG)
- Steps you have followed to attempt to solve the problem

# Index

## Symbols

$ORACLE_HOME, A-5

## A

accessing collected data, 1-4
advanced scheduling options, 3-13
ALL event set, 1-4, 3-5, A-4
Application Programming Interface (API), 1-1
ApplReg event, B-1
Average Elapsed Time data view, 4-8

## B

Basic Statistics for Parse/Execute/Fetch drill-down data view, 4-17
binary files, 1-7, 3-22, A-6
  formatting, 1-4

## C

CACHEIO event set, 3-5, A-4
.cdf files
  *See* collection definition files (.cdf)
client
  required components, 1-5
client/server environment, 1-5
collect.dat file, 1-8
collection definition files (.cdf), 1-6, A-6
  defining length of, 3-12
  deleting, 2-8, 2-11, 3-25
  naming, 3-10
collection entries
  deleting, 3-24
Collection menu, 2-10
collection property pages, 3-19
Collection Wizard, 2-10, 3-2
  creating collections using, 3-1
collections
  changing options, 3-19
  creating, 3-1
  creating like, 3-22
  definition of, 1-2
  deleting, 2-10
  displaying general information about, 3-20
  duration of, 2-9
  formatting to database, 2-10, 3-16
  managing, 1-2
  scheduling, 3-12
  stopping, 3-22
collections in Oracle Trace, A-3
Commit interval, 3-24
Connection event, B-1
control files, 1-8
CPU Statistics data view, 4-9
CPU Statistics for Parse/Execute/Fetch drill-down data view, 4-17
creating collections
  using Oracle Trace Collection Wizard, 3-1
creating like collections, 3-22
cross-facility 3 event, B-4
cross-product items
  *See also* cross-facility 3 event, B-3

## D

data collection files (.dat), 3-25, A-6

deleting, 2-8, 2-11
Data View wizard
choosing an event, 4-20
choosing items, 4-21
choosing sort criteria, 4-23
creating data views using, 4-19
defining number of rows, 4-23
Data Viewer, 4-1
computing statistics, 4-5
starting, 4-2
tips on using
collect data for specific wait events, C-15
usage scenarios, 4-2
data views
creating, 4-13
definition of, 4-1
definitions, 4-10
displaying, 4-5
drill down to related events, 4-14
drill-down
selecting, 4-17
modifying, 4-13
descriptions, 4-14
options, 4-14
sort order, 4-14
options, 4-10
saving, 4-19
*See also* drill-down data views, 4-1
SQL statement, 4-12
statistics, 4-11, 4-13
viewing data, 4-11
data views in Oracle Trace
Average Elapsed Time, 4-8
CPU Statistics, 4-9
Disk Reads, 4-7
Disk Reads/Execution Ratio, 4-7
Disk Reads/Logical Reads Ratio, 4-8
Disk Reads/Rows Fetched Ratio, 4-7
Execute Elapsed Time, 4-9
Fetch Elapsed Time, 4-9
Logical Reads, 4-7
Logical Reads/Rows Fetched Ratio, 4-7
Number of Rows Processed, 4-9
Parse Elapsed Time, 4-9
Parse/Execution Ratio, 4-8
Re-Parse Frequency, 4-8
Rows Fetched/Fetch Count Ratio, 4-9
Rows Sorted, 4-10
Sorts in Memory, 4-10
Sorts on Disk, 4-10
Total Elapsed Time, 4-8
Waits by Average Wait Time, 4-10
Waits by Event Frequency, 4-10
Waits by Total Wait Time, 4-10
database
formatting collection data to, 2-10, 3-16
number of inserts made into, 3-24
DBMS_ORACLE_TRACE_AGENT
stored procedure, 1-7
DEFAULT event set, 3-5, A-4
deleting collection files, 3-25
deleting formatted data, 3-25
Detail report, 1-5
Disconnect event, B-1
discovery process, 1-6
starting from Product menu, 2-11
Disk Reads data view, 4-7
Disk Reads/Execution Ratio data view, 4-7
Disk Reads/Logical Reads Ratio data view, 4-8
Disk Reads/Rows Fetched Ratio data view, 4-7
drill-down
of data views, 4-14
drill-down data views
descriptions, 4-15
modifying, 4-18
selecting, 4-17
SQL statement, 4-16
statistics, 4-16
drill-down data views in Oracle Trace
Basic Statistics for Parse/Execute/Fetch, 4-17
CPU Statistics for Parse/Execute/Fetch, 4-17
I/O Statistics for Parse/Execute/Fetch, 4-17
Parse Statistics, 4-17
Row Statistics for Execute/Fetch, 4-17
duration events, 1-3, B-1
dynamic link library file, 1-8

## E

Edit menu, 2-9

EPC_ERROR.LOG file, C-23
errors
 during discovery, 2-2
ErrorStack event, B-1
event sets, 1-4, A-4
 in Collection Wizard, 3-4
events, 1-3
 server, B-1
Execute Elapsed Time data view, 4-9
Execute event, B-2
exiting Oracle Trace, 2-9
EXPERT event set, 3-5, A-4

## F

facililty.dat files
 *See also* process.dat files, 1-8
.fdf files *See* product definition files (.fdf)
Fetch Elapsed Time data view, 4-9
Fetch event, B-2
File menu, 2-9
FORMAT command in Oracle Trace, A-1, A-2
format history, 3-21
formatter tables
 creating, 1-7
formatting collections, 3-26
formatting data, 3-16
formatting options
 in Collection Wizard, 3-17

## G

General page, 3-20

## H

help
 displaying, 2-11

## I

initialization parameters, 1-3, A-3
INIT.ORA file
 ORACLE_TRACE_ENABLE parameter, 1-7, C-16
INPUT_IO item, B-3
installed products
 listing, 3-4
instrumentation, 1-1
 of Oracle Server, B-1
I/O Statistics for Parse/Execute/Fetch drill-down data view, 4-17
items, 1-3
 cross-product, B-3
 standard resource utilization, B-2
 types of, B-2

## L

Logical Reads data view, 4-7
Logical Reads/Rows Fetched Ratio data view, 4-7
LogicalTX event, B-2

## M

managing collections, 1-2
MAXRS_SIZE item, B-3
menus
 Collection, 2-10
 Edit, 2-9
 File, 2-9
 Help, 2-11
 Product, 2-11
 View, 2-9
Migration event, B-1

## N

node
 preferred credentials for, 3-24
Number of Rows Processed data view, 4-9

## O

OEM
 *See* Oracle Enterprise Manager (OEM)
Oracle Enterprise Manager, 1-2
 starting Oracle Trace from, 2-1
Oracle Intelligent Agent, 1-6
Oracle Server events, B-1

Oracle Trace
  collections, A-3
  command-line interface, A-1
  data views
    Average Elapsed Time, 4-8
    CPU Statistics, 4-9
    Disk Reads, 4-7
    Disk Reads/Execution Ratio, 4-7
    Disk Reads/Logical Reads Ratio, 4-8
    Disk Reads/Rows Fetched Ratio, 4-7
    Execute Elapsed Time, 4-9
    Fetch Elapsed Time, 4-9
    Logical Reads, 4-7
    Logical Reads/Rows Fetched Ratio, 4-7
    Number of Rows Processed, 4-9
    Parse Elapsed Time, 4-9
    Parse/Execution Ratio, 4-8
    Re-Parse Frequency, 4-8
    Rows Fetched/Fetch Count Ratio, 4-9
    Rows Sorted, 4-10
    Sorts in Memory, 4-10
    Sorts on Disk, 4-10
    Total Elapsed Time, 4-8
    Waits by Average Wait Time, 4-10
    Waits by Event Frequency, 4-10
    Waits by Total Wait Time, 4-10
  deleting files, A-3
  drill-down data views
    Basic Statistics for Parse/Execute/Fetch, 4-17
    CPU Statistics for Parse/Execute/Fetch, 4-17
    I/O Statistics for Parse/Execute/Fetch, 4-17
    Parse Statistics, 4-17
    Row Statistics for Execute/Fetch, 4-17
  FORMAT command, A-1, A-2
  formatting data, A-7
  reporting utility, A-9
  START command, A-1
  STOP command, A-1, A-2
Oracle Trace Collection Services, 1-6
Oracle Trace Data Viewer
  *See* Data Viewer, 4-1
Oracle Trace Manager, 1-1, A-4
ORACLE_TRACE_COLLECTION_NAME parameter, A-3, A-4
ORACLE_TRACE_COLLECTION_PATH parameter, A-3
ORACLE_TRACE_COLLECTION_SIZE parameter, A-3
ORACLE_TRACE_ENABLE parameter, 1-7, A-3, C-16
ORACLE_TRACE_FACILITY_NAME parameter, A-3, A-4
ORACLE_TRACE_FACILITY_PATH parameter, A-3
otracexx.dll file, 1-8
otrcfmt command, A-7
otrcfmtc.sql script, A-7
otrcsvr.sql script, 1-7
Output page, 3-21
OUTPUT_IO item, B-3

## P

PAGEFAULT_IO item, B-3
PAGEFAULTS item, B-3
Parse Elapsed Time data view, 4-9
Parse event, B-2
Parse Statistics drill-down data view, 4-17
Parse/Execution Ratio data view, 4-8
PhysicalTX event, B-2
point events, 1-3, B-1
preferences
  defining, 3-23
Preferred credentials
  identifying default format using, 3-23
process.dat files, 1-8
product definition files (.fdf), 1-4
  creating, 2-11
  editing, 2-11
Product menu, 2-11
Progress page, 3-22
property pages, 3-19

## R

regid.dat file, 1-8
Re-Parse Frequency data view, 4-8
repository, 1-6
Row Statistics for Execute/Fetch drill-down data

views, 4-17
Rows Fetched / Fetch Count Ratio data view, 4-9
Rows Sorted data view, 4-10
RowSource event, 1-4, B-1

## S

Schedule page, 3-21
scheduling a collection, 3-12
SCPU item, B-2
scripts
creating stored procedure packages, 1-7
server node
in Collection Wizard, 3-3
required components, 1-6
Sorts in Memory data view, 4-10
Sorts on Disk data view, 4-10
SQL Parse event, 1-3
SQL statements
in data views, 4-12
of drill-down data views, 4-16
SQLSegment event, B-1
standard resource utilization items, B-2
START command in Oracle Trace, A-1
starting Data Viewer, 4-2
from Oracle Enterprise Manager Program Group, 4-3
from within Oracle Trace, 4-3
starting Oracle Trace, 2-1
statistics
computing using Data Viewer, 4-5
of data views, 4-11, 4-13
of drill-down data views, 4-16
status messages, 2-10
STOP command in Oracle Trace, A-1, A-2
stopping a collection, 3-22
stored procedures
DBMS_ORACLE_TRACE_AGENT, 1-7
DBMS_ORACLE_TRACE_USER, 1-7
SUMMARY event set, 3-5, A-4

## T

tables
*See* formatter tables
time zone, 3-15
toolbar
enabling or disabling display, 2-10
icons, 2-7
Total Elapsed Time data view, 4-8
TRACESVR account, 1-7
troubleshooting
conditions to check, A-5

## U

UCPU item, B-2
usage scenarios
for Data Viewer, 4-2

## V

View menu, 2-9

## W

Wait events, B-1
Waits by Average Wait Time data view, 4-10
Waits by Event Frequency data view, 4-10
Waits by Total Wait Time data view, 4-10
wizard
*See* Collection Wizard
*See* Data View wizard, 4-19