Oracle Trace™

Getting Started

Version 2.2

Part No. A38160-1

ORACLE®

Oracle Trace Getting Started

Version 2.2

Part No. A38160-1

Contents

Examples

Figures

Send Us Your Comments

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.

You can send comments to us in the following ways:

- electronic mail — nedc_doc@us.oracle.com
- FAX — 603-897-3334 Attn: Oracle Trace Documentation
- postal service

  Oracle Corporation
  Oracle Trace Documentation
  One Oracle Drive
  Nashua, NH 03062
  USA

If you like, you can use the following questionnaire to give us feedback.

Name _____ Title _____

Company _____ Department _____

Mailing Address _____ Telephone Number _____

Book Title _____ Version Number _____

- 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).

Preface

This manual introduces you to Oracle Trace for OpenVMS Version 2.2. If you have Oracle Trace installed, you can go through the examples in this manual, using the sample application provided.

Intended Audience

This manual is intended for anyone interested in learning more about Oracle Trace for OpenVMS and how to use it to improve an application's performance.

To use Oracle Trace effectively, you should be familiar enough with the applications for which you are collecting data to be able to make sense of the data and use it to the best advantage.

Document Structure

This manual contains the following chapters:

- Chapter 1 provides a high-level description of Oracle Trace and the benefits of using it.
- Chapter 2 builds on your existing knowledge of Oracle Trace and provides information on how to begin using it through the use of a sample application.
- Chapter 3 describes the Oracle Trace Monitor. The Monitor has been a part of Oracle Trace since Version 2.0, but many people were unaware that it was installed on their system.
- Chapter 4 provides examples of the types of reports you can produce with Oracle Trace data.
- Chapter 5 describes how to use PC applications such as Microsoft Excel to analyze Oracle Trace data.
- The Glossary provides definitions of basic Oracle Trace terms used in this manual.

Associated Documentation

The other manuals in the Oracle Trace documentation set are:

- *Oracle Trace Installation Guide* Version 2.2—Provides instructions for installing the Oracle Trace software on an OpenVMS system.
- *Oracle Trace Monitor User's Guide* Version 2.2—Provides instructions for using the Oracle Trace Monitor in more detail than described in this manual.
- *Oracle Trace Collector User's Guide* Version 2.2—Describes how to collect event-based data and instrument applications for collection.
- *Oracle Trace Reporter User's Guide* Version 2.2—Describes how to format and generate hardcopy reports from event-based data.

1
What Is Oracle Trace?

Oracle Trace for OpenVMS is a layered product that monitors performance data for any application—most notably, transaction processing and database applications. It monitors performance by gathering and reporting event-based data from any combination of OpenVMS layered products and application programs containing Oracle Trace service routine calls. Oracle Trace is designed to operate with minimal performance impact on the system and can be used in both development and production environments. In fact, it has been successfully running in production environments since 1990.

1.1 What Is Event-Based Data Collection?

Oracle Trace differs from other collector software in that it is event based, whereas most other collectors are timer based. Timer-based collectors gather data at specified time intervals, at random places within your code. An event-based collector gathers data at predefined locations in your program code when that code is executed.

An Oracle Trace event is an application-defined occurrence. There are two kinds of events:

- Duration events have a separate start and end point. For example, if you were using an automatic teller machine (ATM) to make a withdrawal, it would be a duration event because there is a definite start and end point.
- Point events simply occur, such as when you make an error while using an ATM. There is no separate start and end time.

The advantage of event-based collectors is that you can determine the actual frequency of the execution of events, rather than an average or estimated frequency. Also, event-based collectors give you the ability to collect and report on the resources used by specific events in an application.

1.2 Who Uses Oracle Trace and Why?

Oracle Trace users include application developers, application performance analysts, database administrators, system managers, and capacity planners. They use Oracle Trace to assist them in pinpointing the reasons for an application's poor performance. General reasons for poor performance can be any of the following:

- data access contention
- poorly or incorrectly designed databases
- not enough servers to handle user requests
- inefficient database queries
- actual use of the application differs from the intended use
- inadequate hardware resources

Finding specific causes for these general problems requires data about the application's resource use and response time. Oracle Trace collects a variety of such data from multiple layers of an application. The data is gathered behind the scenes with little impact on the user and minimal impact on system performance.

The following list describes some of the ways you can use Oracle Trace data collection to improve your application's performance:

- Directly relate database transactions and requests to higher level application routines. To achieve optimal application performance you need to know exactly where the cause of poor performance lies. A breakdown of processing and response time, within the application and the underlying database system, shows the root of poor performance.
- Produce reports that contain the data most meaningful to you.
- Tune complex database applications. Because of the increased use of fourth-generation languages (4GLs), you might not even know the underlying query that is returning your data. The tracing mechanism provided by Oracle Trace lets you determine exactly what functions are being performed by multiple users operating against a common database at any given time.
- Collect data from all layers of an application, the user interface, the processing engine, and the database. Oracle Trace is unique in that it can collect information from each of these layers, transcending the proprietary and industry standard application programming interfaces (APIs). Each layer that logs Oracle Trace information can be tied to the layer above

it, which allows you to track a business function throughout its lifetime. The higher the layer, the more important the data because it more closely resembles what your application's user sees.

1.3 Take Advantage of What You Already Have

Many products come ready to work with Oracle Trace without any application modifications. You may already be able to take advantage of the information available using Oracle Trace. For example, you might have a transaction processing application that uses a forms-based user interface to access a database. If this application uses Oracle Rdb ™ or Digital Equipment Corporation's DECforms or ACMS products, you are using products that come ready to work with Oracle Trace.

Several third-party 4GL products, such as SmartStar and NOMAD, contain calls to Oracle Trace routines. This allows their customers to take advantage of Oracle Trace and Oracle Expert™ for Rdb, a knowledge-based system that can use Oracle Trace data.

Note

Whether you have Oracle Trace installed or not, you can learn what products on your system contain Oracle Trace calls by entering the following command:

```
$ LIBRARY/LIST SYS$SHARE:EPC$FACILITY.TLB
```

This command displays a text library directory that contains a list of the Oracle Trace facilities on your system.

1.4 What Are the Oracle Trace Components?

When talking about Oracle Trace, it is important to understand the terms facility and instrumenting. Any product or application program that contains Oracle Trace service routines is called a facility. For example, Oracle Rdb contains Oracle Trace service routines and is therefore considered a facility.

The process of adding Oracle Trace service routines to an application is called instrumenting.

Oracle Trace contains the following components:

- Collector—Gathers event-based data from facilities running on your system and writes the data to one or more collection files.

- Formatter—Converts the raw Oracle Trace data from one or more collection files and writes it to either an Oracle Rdb database or to an OpenVMS RMS file.
- Reporter—Generates tabular reports from data in the Oracle Rdb database.
- Monitor—Displays Oracle Trace data in an environment based on the Motif interface. With the Monitor, you can evaluate performance characteristics of your application as it executes, set thresholds to monitor boundary conditions, and play back displays for subsequent analysis.

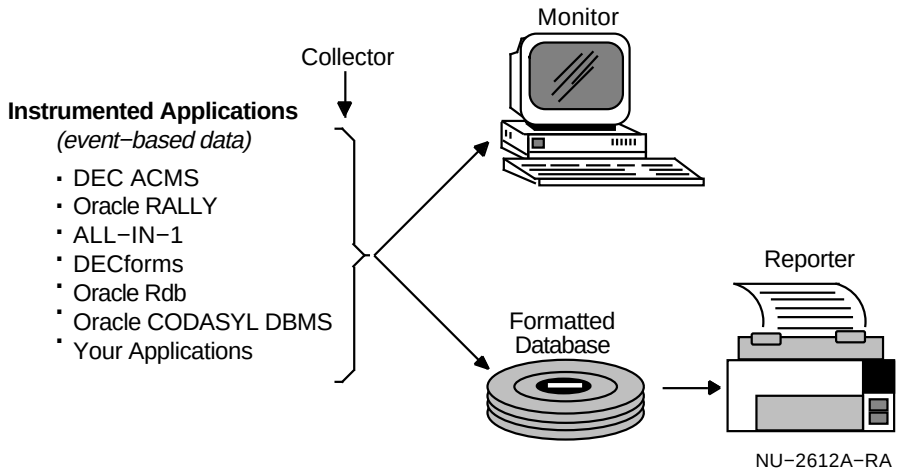
Oracle Trace also uses two databases:

- Administration database—An Oracle Rdb database that is created when you first install Oracle Trace. It contains information about the facilities on your system, the characteristics of the collections you specify, and collection schedules. There is one Administration database per system or cluster.

  Oracle Corporation provides facility definitions for Oracle products that are instrumented for Oracle Trace, such as Oracle Rdb. Facility definitions are also provided for the Digital Equipment Corporation products ALL–IN–1, DEC ACMS, and DECforms. If you are instrumenting your own application, you must create a facility definition for it and insert it into the Administration database.
- History database—An Oracle Rdb database that contains a historical log of Oracle Trace usage. It contains all of the informational and error messages associated with data collection. There is one History database per system or cluster.

Figure 1–1 illustrates the relationship of the various Oracle Trace components.

Figure 1–1 Oracle Trace Components



2
Setting Up a Collection

The purpose of the next few chapters is to acquaint you with Oracle Trace by guiding you through the process of collecting, monitoring, and reporting on data from a sample application provided with Oracle Trace. The application simulates a simple bank automated teller machine (ATM).

Before describing how to set up a collection, this chapter shows you how to invoke Oracle Trace and how to verify that it is running on your system. You are also given the chance to run the sample application so that you can become familiar with it, before proceeding with setting up a collection.

2.1 Invoke Oracle Trace

Oracle Trace provides a command-line interface, which you can use in two ways:

1. Enter the Oracle Trace environment by issuing the COLLECT command at the DCL prompt. Oracle Trace prompts you for commands until you issue the EXIT command to return to DCL level. For example,

   ```
   $ COLLECT
   Trace> SHOW VERSION
   Oracle Trace Version 2.2
   Trace> EXIT
   $
   ```

   Working within the Oracle Trace environment results in better performance.

2. Issue Oracle Trace commands at the DCL prompt, by prefacing them with the keyword COLLECT. This allows you to remain at DCL level while using Oracle Trace. For example,

   ```
   $ COLLECT SHOW VERSION
   Oracle Trace Version 2.2
   $
   ```

2.2 Verify Oracle Trace Is Running

If you would like to try out the sample application before proceeding with setting up a collection, first verify that Oracle Trace is running on your system. To do this, enter the following command:

```
$ SHOW SYSTEM
```

If Oracle Trace is running, you will see EPC$REGISTRAR listed as one of the processes. The Registrar handles all communication between the applications running on your system and the Oracle Trace software. The Registrar tells Oracle Trace to start or stop collecting data and maintains a list of the facilities available for data collection.

```
VAX/VMS V5.5-2 on node OOTOOL 30-MAR-1995 10:55:07.82 Uptime 28 19:57:20
  Pid    Process Name   State Pri    I/O      CPU      Page flts Ph.Mem

   .
   .
   .
2D402EE4 SYSTEM         LEF     9   8261  0 00:01:25.40    17548    895
2D4016E7 EPC$REGISTRAR  HIB    10   2256  0 00:00:20.98     5952   5309
   .
   .
   .
```

If Oracle Trace is not running, you must ask your system manager to execute the Oracle Trace startup procedure. If you have the necessary system privileges (as described in *Oracle Trace Collector User's Guide*), you can execute the startup procedure yourself by executing the following command:

```
$ @sys$startup:epc$startup
```

2.3 Run the Sample Application

Enter the following command to try out the sample application before setting up a collection:

```
$ RUN EPC$EXAMPLES:EPC$ATM-SAMPLE-EXTENDED.EXE
```

The first time that you run the sample application, it automatically builds a database. This might take a few minutes, during which a message is displayed similar to the following:

```
Unable to bind to ATM database, building a new one.
                 Please wait.
```

The initial ATM screen prompts you for an account number, as shown in the following example. Enter any number between 1 and 10.

```
YOURBANK -- Automatic Teller Machine

 Please enter your account number (1 - 10)
 (Enter 0 to exit the application.)

                 Account number:
```

The sample application then displays a screen with ATM transaction choices, as follows:

```
YOURBANK -- Automatic Teller Machine

                      1.  Withdrawal
                      2.  Deposit
                      3.  Balance

                      4.  Exit

    Choice:
```

Practice using the different choices to familiarize yourself with the sample application, which has been instrumented to gather information about the user interface. Duration events are defined for each of the basic transactions: checking a balance, depositing funds, and withdrawing funds. A point event is defined to note execution of the error-handling procedure. The goal is to be able to answer questions such as:

- How long does it take to complete a transaction?
- Do customers check their balance before or after every transaction?
- Could overdrafts be reduced or eliminated by displaying the balance on the withdrawal display?
- Are ATM machines used primarily for deposits or withdrawals?

The sample application uses the cross-facility feature—it calls Oracle Rdb and has been instrumented so that events in the sample application can be related to the underlying Oracle Rdb events. This means that the report you generate will show the resources that both the application and Oracle Rdb use for each event.

Before you exit the application, it is a good idea to find out what version of Oracle Rdb your process is running because you will need this information to set up a collection. To determine the version, start another session on the same system on which you are running the sample application and issue the following command:

```
$ COLLECT SHOW REGISTER/NOCLUSTER
```

Make a note of the version of Oracle Rdb listed. (The /NOCLUSTER qualifier limits the display to information about your system, rather than the entire cluster.)

The remainder of this chapter describes how to set up a collection, using the following steps:

1. Create facility definitions and insert them in the Administration database.
2. Create a facility selection.
3. Schedule a collection.
4. Start the application for which data is to be collected.

2.4 Create and Insert Facility Definitions

A facility definition is a description of the events and items that Oracle Trace can capture for a particular facility. Oracle Trace cannot collect data for a facility unless there is a facility definition in the Administration database for the facility.

Before you can collect data for facilities, you must verify that the Administration database contains facility definitions for them. If it does not, you must create the necessary facility definitions for them and insert them in the Administration database.

For the purposes of the sample application, Oracle Corporation provides facility definitions for the ATM_SAMPLE and Oracle Rdb facilities. To check for the presence of facility definitions in the Administration database use the SHOW DEFINITION command. The following example of the SHOW DEFINITION command limits the display to the RDBVMS facility.

```
$ COLLECT SHOW DEFINITION RDBVMS
```

If your Administration database contains facility definitions for Oracle Rdb, Oracle Trace lists them in a display similar to the following:

```
19-APR-1995 16:15       Facility Definition Information          Page 1
Names Only Report                                       Oracle Trace V2.2

 Facility:            Version:    Creation Date:      Class:
 -------------------  ----------  ------------------  -----------------
                      V6.0-0      31-JAN-1995 16:16   ALL
                                                      PERFORMANCE       (D)
                                                      PERFORMANCE_NO_CF
                                                      Oracle Expert
                                                      Oracle Expert_NO_CF
                      V5.1-0       5-JAN-1995 16:40   ALL
                                                      PERFORMANCE       (D)
                                                      PERFORMANCE_NO_CF
                                                      Oracle Expert
                                                      Oracle Expert_NO_CF
```

If you receive an error message when you issue the SHOW DEFINITION command, it is probably because your version of Oracle Rdb does not match the version that existed when the Administration database was created. To solve this problem, contact your system manager to find out which version of Oracle Rdb was specified when Oracle Trace was installed. You can then execute the following command procedure, replacing `version_num` with the correct version number:

```
$ @SYS$SHARE:RDBVMS_SETVER version_num
```

To learn more about creating facility definitions and inserting them into the Administration database, refer to the *Oracle Trace Collector User's Guide*.

2.5 Create a Facility Selection

Once you have created the necessary facility definitions (or verified that they already exist), the next step in setting up a collection is to create a facility selection. A facility selection specifies the facilities from which to collect data and the kinds of information to collect.

Since the sample application provided with Oracle Trace calls Oracle Rdb, you need to create a facility selection that specifies Oracle Rdb and the ATM_SAMPLE facility. When you create a selection, you must be sure that the version number you specify for each facility is actually the version of the facility that your process uses. Otherwise, an error results.

Enter a command similar to the following to create a facility selection for Oracle Rdb. Be sure to use the correct version number (not necessarily the one used in the example).

```
$ COLLECT CREATE SELECTION SAMPLE_SELECTION - ❶
_$/OPTIONS ❷
Option> FACILITY RDBVMS /VERSION="V5.1-0" ❸
Option> FACILITY ATM_SAMPLE /VERSION="V1.2-0" ❹

Option> CTRL/Z
%EPC-S-SELCRE, Selection SAMPLE_SELECTION was created
```

❶ Associate the name SAMPLE_SELECTION with the facility selection.

❷ Tell Oracle Trace to prompt interactively for options.

❸ Specify RDBVMS Version 5.1-0 as a facility. Be sure that the version number you specify corresponds to the version of Oracle Rdb that your process runs. If you are using multiversion Oracle Rdb, use the following command to display the currently running version:

```
$ @SYS$SHARE:RDBVMS_SHOVER
```

If you are not using multiversion Oracle Rdb, use the following command to display the currently running version:

```
$ RMU/SHOW VERSION
```

❹ Specify ATM_SAMPLE Version 1.2–0 as a facility.

If you get a message that SAMPLE_SELECTION already exists, or if you just want to verify what it contains, use the following command:

```
$ COLLECT SHOW SELECTION SAMPLE_SELECTION
30-MAR-1995 10:49  Facility Selection Information    Page 1
Oracle Trace V2.2
Selection Name      Facility         Version     Class
------------------  ---------------  ----------  -----------
SAMPLE_SELECTION    ATM_SAMPLE       V1.2-0      ALL
                    RDBVMS           V5.1-0      PERFORMANCE
```

The SHOW SELECTION command displays the name of the selection, the facilities and their versions, and the class of data Oracle Trace is collecting. A class is a user-defined group of data related to a specific purpose, such as performance or debugging.

2.6 Schedule a Collection

Enter a command similar to the following to begin a 30-minute collection:

```
$ COLLECT SCHEDULE COLLECTION SAMPLE_COLLECTION - ❶
disk:[directory]SAMPLE_DATA.DAT - ❷
_$ /SELECTION=SAMPLE_SELECTION - ❸
_$ /DURATION=:30 - ❹
-$ /FLUSH_INTERVAL=(00:00:02) - ❺
_$ /NOCLUSTER - ❻
-$ /COLLECTION_FILES=(PROTECTION=(W:RW)) ❼
%EPC-S-SCHED, Data collection SAMPLE_COLLECTION is scheduled
```

❶ Associate the name SAMPLE_COLLECTION with the collection.

❷ Specify the disk and directory you want to receive the collection data. If you specify a file name without a device and directory specification, Oracle Trace creates the file in your current directory. In this example, the collection file is named SAMPLE_DATA.DAT.

❸ Associate the selection SAMPLE_SELECTION with this collection.

❹ Specify a 30-minute collection time, starting when you execute this command.

❺ Specifies the interval in seconds for Oracle Trace to write out all process buffers to the data collection file. It is critical that you specify a flush interval if you intend to monitor your collection. Otherwise, Oracle Trace uses the default flush interval which has a very large buffer size. As a result, the buffer would probably never be full enough for you to see anything displayed in the Monitor. Oracle Corporation recommends a flush interval of 1 or 2 seconds.

❻ Limit the collection to your system.

❼ Specify a protection of world read and write access for the collection file. You should always do this to ensure that collecting applications have privileges to write to the collection file.

You can use the following command to verify the collection you scheduled:

```
$ COLLECT SHOW COLLECTION SAMPLE_COLLECTION
30-MAR-1995 10:56     Scheduled Collections                  Page 1
Brief Report                                        Oracle Trace V2.2

Collections scheduled for node ALICIA
Collection Name  Selection Name  Start             End
---------------- --------------- ----------------- -----------------
SAMPLE           SAMPLE          30-MAR-1995 10:55 30-MAR-1995 11:25
_COLLECTION       _SELECTION
```

At this point, you have done everything you need to do to set up the collection. However, although you successfully scheduled the collection and it is running, it cannot collect data for the ATM application until you start the application, as described in the following section.

2.7 Start the Application and Verify that Oracle Trace Is Collecting

Enter the following command to start the sample application:

```
$ RUN EPC$EXAMPLES:EPC$ATM-SAMPLE-EXTENDED.EXE
```

You can verify that Oracle Trace is actively collecting data by starting another session on this node and entering the following command:

```
$ COLLECT SHOW REGISTER/NOCLUSTER
30-MAR-1995 16:35    Register Information for node ALICIA Page 1
Oracle Trace V2.2 Registrations actively collecting
Node: ALICIA Collection: SAMPLE      Selection: SAMPLE_SELECTION
_COLLECTION
Process   Process Name     Facility  Version   Registration Id
--------  --------------  --------  --------  -----------------
-> 2D4028D2  FLANDERS     RDBVMS     V5.1-0
->                           ATM     V1.2-0    ATM APPLICATION EXT
                           _SAMPLE
$255$DUA60:[FLANDERS.ATM_RDB_EXAMPLE]EPC$ATM-SAMPLE-EXTENDED.EXE;19
ADMIN20:[FLANDERS.ATM_RDB_EXAMPLE]SAMPLE_DATA.DAT;13
```

The arrow (->) at the beginning of each line indicates that data collection is taking place for that facility. The Registration Id is a unique identifier associated with the facility.

Continue depositing and withdrawing money to accounts for a few minutes. Keep track of the deposits and withdrawals that you perform so you can see how Oracle Trace reports them.

The collection that you schedule will run for 30 minutes. The collection starts when you enter the command and stops 30 minutes later. If you do not run the sample application during this time, the report will not contain any event data.

2.8 Stop the Collection

If you do not want to wait 30 minutes for the collection you scheduled to complete, you can cancel the collection with the following command:

```
$ COLLECT CANCEL COLLECTION SAMPLE_COLLECTION - ❶
_$ /NOCONFIRM ❷
%EPC_S_SCHED_CANCEL, Collection Sample_collection is set to aborting.
```

❶ Specify which collection to cancel.

❷ Suppress confirmation prompt.

Chapter 3 describes how you can monitor a collection as it takes place, or how you can replay a data collection file at a later time.

3
Monitoring Collections

Oracle Trace provides a Monitor that allows you to watch the collection as it takes place or to play back previously collected data. In either case, the functions of the Monitor allow you to progressively disclose resource use at the following levels:

- Per process
- Per facility within a process
- Per event within a facility
- Grouped by item values within an event

This progressive disclosure allows you to identify performance problems based on your criteria such as response time, I/O rate, or CPU time. You can then determine exactly where the problem is occurring in terms of application-specific work units or data.

The Monitor's ease of use is enhanced by options that allow you to replay collections at various rates of speed and by the option that allows you to customize the display and then save the settings for future sessions.

Adding to the Monitor's usefulness is the option that allows you to select threshold settings and enable notification mechanisms that issue customized alert messages or perform automatic recovery actions tailored to your specific environment.

This chapter briefly introduces you to the Oracle Trace Monitor. For a complete explanation of the Monitor's many capabilities, refer to the description and examples in the *Oracle Trace Monitor User's Guide*.

3.1 Start the Monitor

To monitor the collection you started in Chapter 2, enter the following command:

```
$ COLLECT MONITOR SAMPLE_DATA.DAT
```

If your collection has ended, you can still monitor it by using the /REPLAY qualifier to replay the data collection file:

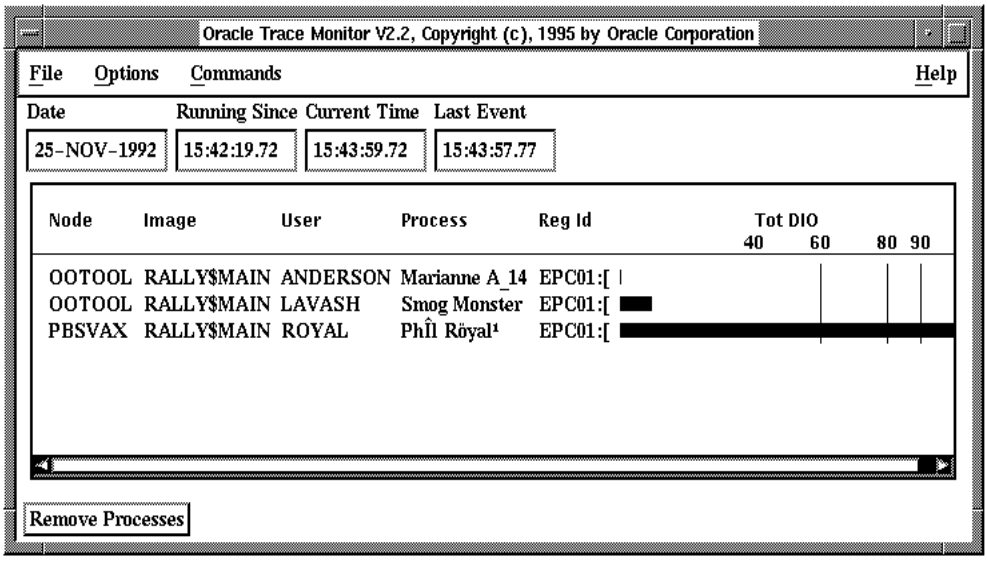
```
$ COLLECT MONITOR /REPLAY SAMPLE_DATA.DAT
```

If you do not have a data collection file of your own to monitor, you can use the Oracle Trace demonstration file by entering the following command:

```
$ COLLECT MONITOR /REPLAY EPC$EXAMPLES:DEMO_DATA.DAT
```

The Monitor displays the Process window, similar to the one shown in Figure 3–1.

Figure 3–1 The Process Window



NU-3124A-RA

The Process window is used to identify problematic processes in general terms. As a default, the Process window displays total direct input/output (DIO) for currently running processes. You can use the choices in the Options menu to change the item displayed. See the *Oracle Trace Monitor User's Guide* for further information.

3.2 Display Data in More Detail

To display resource use for the facilities within a process, double click on a process name in the Process window. This expands the information displayed for the process, by also showing the facilities within the process. For example, if you are monitoring the collection file you created when running the sample ATM application and you double click on your process name, the facilities displayed from your sample collection are ATM_SAMPLE and RDBVMS.

You can then double click on the facility names ATM_SAMPLE and RDBVMS to display specific events within the facility. For example, if you double click on ATM_SAMPLE, events such as BALANCE_EVENT are displayed in a separate window.

From here, you can proceed to associate data items with each event. Data items can be resource use statistics that are standard among applications or specific to a particular application. Data items can also relate events among layered products or applications. Working at this level of detail is explained thoroughly in the *Oracle Trace Monitor User's Guide*.

3.3 Pause the Monitor Display

On occasion, you might want to pause the Monitor to take a closer look at the display. To do this, choose the Pause menu item from the Commands menu in any window.

Choose the Resume menu item from the Commands menu to begin displaying Monitor activity again. It may take a while to "catch up" to real-time viewing when you choose Resume. This is because the Monitor displays, at normal speed, all events that occurred during the Pause interval.

3.4 Set Threshold Values

Threshold values define limits of resource use according to the following levels: Low, Medium, High, Critical, and Severe. You can define the values associated with each level so that they meet the needs of your application.

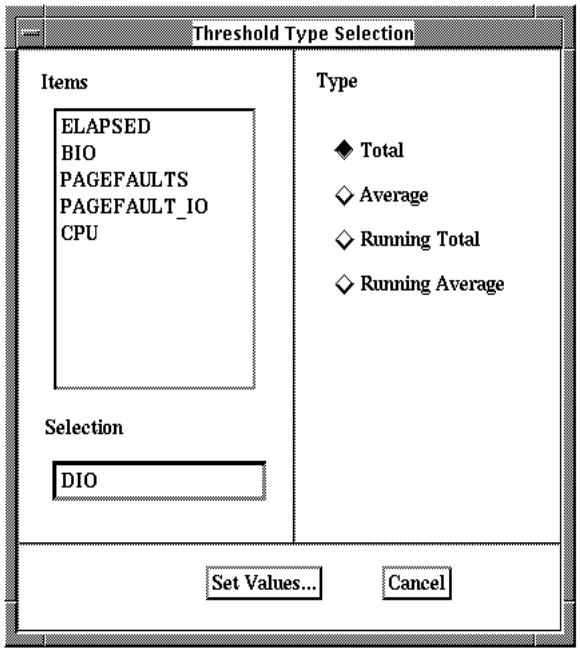
This manual presents the basic steps for working with thresholds to familiarize you with the concept. If you want more specific information, such as item definitions, refer to the *Oracle Trace Monitor User's Guide*.

Follow these steps to set threshold values:

1. Choose the Select Threshold Values menu item from the Options menu.

The Monitor displays the Threshold Type Selection dialog box, as shown in Figure 3–2.

Figure 3–2 Threshold Type Selection Dialog Box



NU–3129A–RA

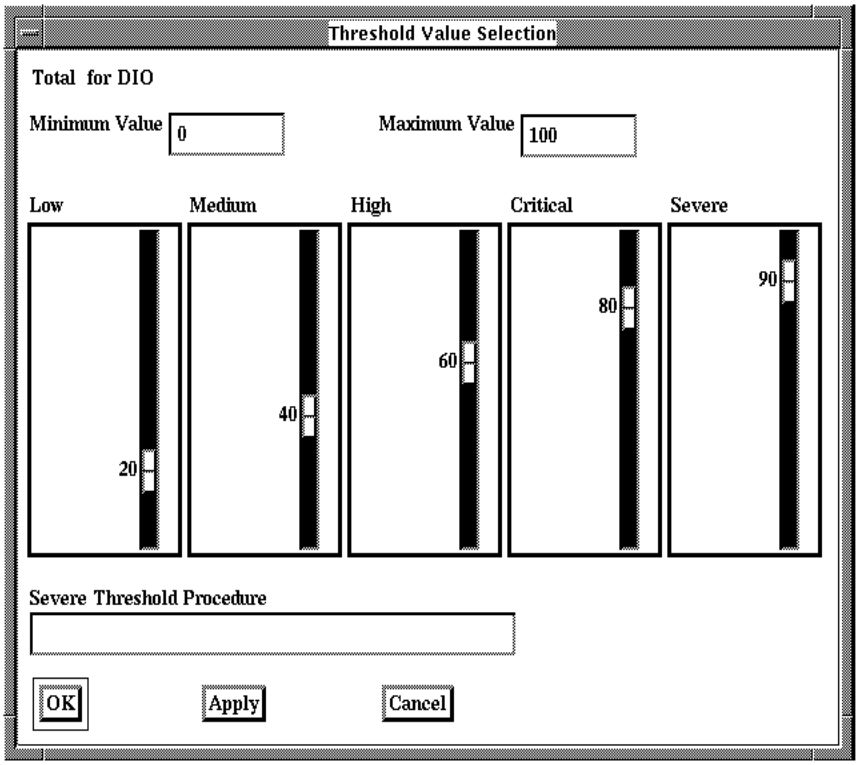
2. Choose a statistic type by clicking on the appropriate toggle button. The available statistic types are defined in the following table:

Statistic Type	Definition
Total	The total count or value in the last update cycle.
Average	The average count or value in the last update cycle.
Running Total	The total count or value since the last reset operation.
Running Average	The average count or value since the last reset operation.

3. Click on a name in the Items list and click again in the Selection text entry box to place it there.
4. Click on the Set Values push button.

The Monitor displays the Threshold Value Selection dialog box, as shown in Figure 3–3.

Figure 3–3 The Threshold Values Selection Dialog Box



NU–3130A–RA

5. Specify minimum and maximum values by entering them in the appropriate fields. Specify Low, Medium, High, Critical, and Severe values by positioning the sliders appropriately.
6. If you have a command procedure that you want executed when severe thresholds are reached for the statistic/item pair, enter a file specification for the procedure in the Severe Threshold Procedure text entry box.
7. At this point you can still click on the Cancel button to cancel any changes you have made and close the Threshold Value Selection dialog box. Or you can apply the changes by proceeding to the next step.

8. Click on the OK or the Apply button to see the new thresholds reflected in the bar graph. The Apply button continues to display the Threshold Value Selection dialog box; the OK button removes it.

9. Now you can either set threshold values for another statistic/item pair or you can finish setting threshold values.

   To set values for another pair, return to the Threshold Type Selection dialog box (step 2).

   To finish setting threshold values, remove the Threshold Value Selection dialog box (if it is still open) by clicking on the OK button. Then click on the Cancel button in the Threshold Type Selection dialog box to remove that dialog box.

10. *Optional step:* You can save the threshold values you set as the defaults by choosing the Save Current Settings menu item from the Options menu.

3.5 Stop the Monitor

To stop the Monitor, choose the EXIT menu item from the File menu at the top of the Process window. If you stop the Monitor before the specified data collection period expires, data collection continues even after the Monitor is stopped.

4

Producing Oracle Trace Reports

Before you can produce a report, you must format the data in the data collection file by using the Oracle Trace FORMAT and REPORT commands.

The examples in this chapter use the Oracle Trace demonstration data file. If you want to use your own collection file, specify the name of your .DAT file in place of EPC$EXAMPLES:DEMO_DATA.DAT.

Use the following FORMAT command to format the sample data that you collected:

```
$ COLLECT FORMAT EPC$EXAMPLES:DEMO_DATA.DAT - ❶
_$ FORMATTED_SAMPLE.RDB ❷
%EPC-I-FMT_RDB_CREATE, Creating database FORMATTED_SAMPLE.RDB
%EPC-S-FMT_RDB_SUCCESS, Successfully created database
%EPC-I-FMT_DCF_BEGIN, Formatting data file EPC$EXAMPLES:DEMO_DATA.DAT
%EPC-S-FMT_DCF_SUCCESS, Successfully formatted data file EPC$EXAMPLES:
     DEMO_DATA.DAT
%EPC-S-FMT_SUCCESS, Formatting successfully completed
```

❶ Specify EPC$EXAMPLES:DEMO_DATA.DAT as the collection file to be formatted.

❷ Specify FORMATTED_SAMPLE.RDB as the name of the formatted database to create.

Once the data is formatted, you can use the following command to generate a report:

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB - ❶
_$ /TYPE=SUMMARY - ❷
_$ /STATISTICS=ALL - ❸
_$ /FACILITY=(RDBVMS) - ❹
_$ /OUTPUT=SAMPLE_REPORT.TXT ❺
%EPC-S-RPCL_SUCCESS, Report successfully completed
$
```

❶ Specify FORMATTED_SAMPLE.RDB as the database.

❷ Specify a summary report.

❸ Request a report for all statistics.

❹ Specify RDBVMS as the facility.

❺ Specify SAMPLE_REPORT.TXT as the report file.

To learn more about producing Oracle Trace reports, refer to the *Oracle Trace Reporter User's Guide.*

Note

Oracle Trace does not attempt to analyze or modify the performance of an application or database. Its function is to collect data requested by users and to provide tabular reports based on that data. Interpreting these reports is the responsibility of the user or of other layered products.

The report produced by the sample commands in this chapter is similar to the one shown in Example 4–1.

Example 4–1 Sample Oracle Trace Summary Report

```
5-APR-1995 10:28                Summary Report                        Page 1
Selection: RALLY_RDB_SEL                                    Oracle Trace V2.2

Event:  Database        In Facility:  RDBVMS          Version:  V4.0-2

         Timestamp                  Client PC    Stream Id

Minimum  25-NOV-1994 15:43:08.33         693F            1
Maximum  25-NOV-1994 15:43:57.65         6D94            2
Mean                                   6B69.00         1.50
Std Dev                                 25F.00         0.54
95 Prct                                7010.00         2.57
Total                                    28479            9
Count    6

%EPC-I-RPQU_BAD_95, 95 Prct for events with counts under 1000 are less precise


5-APR-1995 10:28                Summary Report                        Page 2
Selection: RALLY_RDB_SEL                                    Oracle Trace V2.2

Event:  Request Actual  In Facility:  RDBVMS          Version:  V4.0-2

           Elapsed     AIJ File  BUFFERED IO       Buffer    Client PC
                         Writes                  Pool Rds
```

(continued on next page)

Example 4–1 (Cont.) Sample Oracle Trace Summary Report

```
Minimum       0.00           0           0           0        42D4
Maximum     111.67           0         345         108        7AE6
Mean          2.13        0.00        3.85       10.55     7740.00
Std Dev      10.46        0.00       20.57       29.06      6CF.00
95 Prct      22.63        0.00       44.17       67.53     8499.00
Total      1589.00           0        2873        7864     15B0947
Count          745

            Comp Sts     CPU TIME  CURREN    Data File    Data File
                                   T PRIO        Reads         Wrts

Minimum            1            0       3            0            0
Maximum            1          252       9           99            2
Mean            1.00        13.14    4.23         7.74         0.00
Std Dev         0.00        34.89    1.20        21.44         0.10
95 Prct         1.00        81.53    6.58        49.78         0.20
Total            745         9791    3152         5767            4
Count            745

           DIRECT IO      Free VM       Get VM       Locks        Locks
                            Bytes        Bytes    Released    Requested

Minimum            0            0            0           0            0
Maximum          174        76944       132360          83          229
Mean            9.94      2599.24      3005.65        2.73        11.06
Std Dev        26.94      7359.64     10875.30        8.17        30.50
95 Prct        62.75     17024.14     24321.25       18.76        70.85
Total           7412      1936440      2239216        2038         8242
Count            745

          Lock Stall   PAGEFAULTS    PAGEFAULT         Prom   Req        Rqsts
               Time                       IOs    Deadlocks  Oper     Deadlock

Minimum            0            0            0           0     0            0
Maximum          135          635            0           0     0            0
Mean            0.48         6.17         0.00        0.00  0.00         0.00
Std Dev         5.65        46.93         0.00        0.00  0.00         0.00
95 Prct        11.57        98.16         0.00        0.00  0.00         0.00
Total            364         4598            0           0     0            0
Count            745

            Req Hndl    Rqsts Not       Rqsts    Root File    Root File
                             Q'd      Stalled       Reads         Wrts

Minimum            0            0            0           0            0
Maximum            4          141           26           1            0

5-APR-1995 10:28               Summary Report                      Page 3
Selection: RALLY_RDB_SEL                                  Oracle Trace V2.2
```

(continued on next page)

Example 4–1 (Cont.) Sample Oracle Trace Summary Report

```
            Req Hndl    Rqsts Not        Rqsts    Root File    Root File
                             Q'd      Stalled       Reads        Wrts

Mean            1.03         2.65         0.24         0.00         0.00
Std Dev         0.60         9.16         1.70         0.05         0.00
95 Prct         2.20        20.61         3.59         0.10         0.00
Total            768         1979          186            2            0
Count            745

            RUJ File     RUJ File    Stream Id    Trans Seq      VIRTUAL
               Reads       Writes                       Num         SIZE

Minimum            0            0            1          472        10896
Maximum            0            0            2          525        13478
Mean            0.00         0.00         1.10       493.39     12336.83
Std Dev         0.00         0.00         0.30        14.30       612.05
95 Prct         0.00         0.00         1.69       521.43     13536.46
Total              0            0          820       367582      9190940
Count            745

           GLOBAL WS   PRIVATE WS      WORKING
                                      SET SIZ

Minimum          877         1671         8192
Maximum         1377         2565        15000
Mean         1288.31      2359.47     10440.01
Std Dev        62.12       126.49      3203.85
95 Prct      1410.07      2607.41     16719.56
Total         959791      1757812      7777808
Count            745

%EPC-I-RPQU_BAD_95, 95 Prct for events with counts under 1000 are less precise

5-APR-1995 10:28                Summary Report                        Page 4
Selection: RALLY_RDB_SEL                                   Oracle Trace V2.2

Event:  Transaction     In Facility:  RDBVMS          Version:  V4.0-2

            Elapsed     AIJ File  BUFFERED IO       Buffer    Client PC
                          Writes                  Pool Rds

Minimum        0.09            0            0            1         42D4
Maximum      111.68            0          345          130         7AE6
Mean          11.97         0.00        24.50        61.79      6CF7.00
Std Dev       23.57         0.00        50.23        46.91       75E.00
95 Prct       58.18         0.00       122.97       153.74      7B68.00
Total       1581.19            0         3235         8157       382FAD
Count           132
```

(continued on next page)

Example 4–1 (Cont.) Sample Oracle Trace Summary Report

```
            CPU TIME    CROSS FAC  CROSS FAC 2  CROSS FAC 3  CROSS FAC 7
                                14

Minimum            0            0            0            0            0
Maximum          363            0            0            0            0
Mean           73.70         0.00         0.00         0.00         0.00
Std Dev        74.38         0.00         0.00         0.00         0.00
95 Prct       219.50         0.00         0.00         0.00         0.00
Total            9729            0            0            0            0
Count             132

          CURREN    Data File    Data File    DIRECT IO      Free VM
          T PRIO         Reads         Wrts                   Bytes

Minimum        3             1            0            2            0
Maximum        9            99            2          174       100064
Mean        5.20         45.28         0.03        55.34     24772.72
Std Dev     1.78         34.22         0.24        43.24     20264.02
95 Prct     8.71        112.36         0.51       140.10     64490.21
Total        687          5978            4         7305      3270000
Count        132

              Get VM  Lock        Locks        Locks   Lock Stall   PAGEFAULTS
               Bytes  Mode     Released    Requested         Time

Minimum            0     8            0            3            0            0
Maximum       202480     9           83          229          136          635
Mean        36785.51  8.39        18.94        67.75         2.80        39.85
Std Dev     34760.18  0.49        18.40        47.91        13.32       118.82
95 Prct    104915.46  9.35        55.02       161.68        28.91       272.75
Total        4855688  1108         2501         8944          370         5261
Count            132

            PAGEFAULT         Prom        Rqsts    Rqsts Not        Rqsts
                  IOs    Deadlocks     Deadlock         Q'd      Stalled

Minimum            0            0            0            0            0
Maximum            1            0            0          141           26
```

(continued on next page)

Example 4–1 (Cont.) Sample Oracle Trace Summary Report

```
5-APR-1995 10:28                Summary Report                       Page 5
Selection: RALLY_RDB_SEL                                  Oracle Trace V2.2

           PAGEFAULT        Prom       Rqsts    Rqsts Not       Rqsts
                 IOs   Deadlocks    Deadlock          Q'd     Stalled

Mean            0.00        0.00        0.00        15.26        1.57
Std Dev         0.08        0.00        0.00        17.34        4.17
95 Prct         0.17        0.00        0.00        49.25        9.74
Total              1           0           0         2015         208
Count            132

           Root File    Root File    RUJ File    RUJ File   Stream Id
               Reads        Wrts       Reads      Writes

Minimum            0           0           0           0           1
Maximum            1           0           0           0           2
Mean            0.01        0.00        0.00        0.00        1.39
Std Dev         0.12        0.00        0.00        0.00        0.49
95 Prct         0.25        0.00        0.00        0.00        2.35
Total              2           0           0           0         184
Count            132

           Trans Seq      VIRTUAL    GLOBAL WS   PRIVATE WS      WORKING
                 Num         SIZE                               SET SIZ

Minimum          472       10896         895        1676        8192
Maximum          525       13478        1377        2565       15000
Mean          493.18     12293.15     1270.65     2342.27    10358.18
Std Dev        14.88       627.01       83.87      153.28     3183.04
95 Prct       522.35     13522.11     1435.04     2642.71    16596.95
Total          65101      1622696      167727      309180     1367280
Count            132

%EPC-I-RPQU_BAD_95, 95 Prct for events with counts under 1000 are less precise


5-APR-1995 10:28                    Index                            Page 6
Selection: RALLY_RDB_SEL                                  Oracle Trace V2.2

Report Index

Facility Name               Event Name                      Join item  Page

RDBVMS                      Database                                      1
RDBVMS                      Request Actual                                2
RDBVMS                      Transaction                                   4
```

5

Using Your PC to Analyze Oracle Trace Data

You can access an Oracle Trace formatted database from Microsoft Excel or other PC analysis and reporting tools. This functionality is an advanced feature of Oracle Trace, but is well worth pursuing if you are interested in using your PC to analyze Oracle Trace data.

If you have Microsoft Excel Version 5.0, you can use the Open Database Connectivity (ODBC) API to directly access an Oracle Trace formatted database. If you have Microsoft Excel Version 4.0 or earlier, you can use the instructions in this section to access an Oracle Trace formatted database.

The following example command procedures show you how to access the data in an Oracle Trace formatted database for the purpose of analysis using Microsoft Excel.

```
EPC$EXAMPLES:EPC$ACTUALS_VIEW.COM
EPC$EXAMPLES:EPC$ACTUALS_VIEW_TO_CSV.COM
```

The procedure EPC$ACTUALS_VIEW.COM creates a view called DB_ACTUALS using interactive SQL from the base table EPC$1_221_REQUEST_ACTUAL. It attaches to a formatted database pointed to by the logical EPC$FORMAT_DB.

The procedure EPC$ACTUALS_VIEW_TO_CSV.COM generates a comma separated values (CSV) file from the DB_ACTUALS view for input into Microsoft Excel. The name of the CSV file is PCDATA.CSV.

Requirements

The following requirements must be met before you can use these command procedures:

1. The formatted database must have been created using the /ELAPSED_TIME qualifier with the COLLECT FORMAT command, if you want the elapsed time field defined in the resulting DB_ACTUALS view. For example:

```
$ COLLECT FORMAT/ELAPSED_TIME epc$examples:demo_data.dat -
_$ demo_data
```

2. The logical name EPC$FORMAT_DB should point to the name of the formatted database.
3. You must use Oracle Rdb Version 4.2 or higher with the SQL command procedure, because the computed-by logic is not available in earlier Oracle Rdb versions.
4. Before you use EPC$ACTUALS_VIEW_TO_CSV.COM, a view called DB_ACTUALS must reside in the formatted database. If the view is named differently, edit the "from db_actuals..." line in this command procedure. See the procedure EPC$EXAMPLES:EPC$ACTUALS_VIEW.COM for an example of how to define a view.

Glossary

Administration database

A single Oracle Rdb database that contains the Oracle Trace facility definitions, facility selections, and schedule information. There is one Oracle Trace Administration database per system or cluster. You can reference the Administration database with the logical EPC$ADMIN_DB.

application program

A sequence of instructions and routines, not part of the basic operating system, designed to serve the specific needs of a user. An application program can be instrumented with Oracle Trace service routine calls. Also referred to as a **facility**, especially after the Oracle Trace calls have been added.

buffered I/O

Buffered I/O used during processing. This indicates that a device is transferring data to or from a buffer in the nonpaged pool.

collection class

A set (or group) of events and items that can be collected for a facility. Classes for a facility are specified in the facility definition. Users refer to a class when creating a facility selection. There can be one or more classes for each facility. Typical classes include CAPACITY_PLANNING, PERFORMANCE, and WORKLOAD.

collection interval

See **interval**.

collection name

A 1- to 16-character string that represents the name of a particular collection.

CPU usage

CPU time charged to the process.

cross-facility feature

An Oracle Trace feature that allows programmers instrumenting Oracle Trace service routine calls in applications to relate events among one or more facilities. This feature is useful when other methods such as explicitly passing data items through the application programming interface (API) are not possible.

General users with a knowledge of how the facilities they are tracking have implemented the cross-facility feature, can produce reports in which related events are joined, using the /JOIN qualifier of the REPORT command.

data collection

The process of collecting data on a system or cluster. Criteria in the scheduling of data collection include when to collect data, where to put the output, and which facility selection to use. Data collection must be scheduled on a system in order to collect data. Multiple collections may be active on a node or cluster at any time.

data collection file

A file that contains raw data gathered during data collection. A single data collection file stores data for one or more OpenVMS processes. The SCHEDULE COLLECTION command creates the file and the processes running instrumented applications write to the file.

data formatting

Organizing collected data into an Oracle Rdb database or a formatted OpenVMS RMS file. Collected data must be formatted before Oracle Trace can generate reports based on it.

direct I/O

Direct I/O used during processing. In a direct I/O operation, a device transfers directly to or from memory using the users address space.

duration event

See **event**.

elapsed time

Time interval between the start and end of a duration event.

end event

The end of a duration event. See also **event**.

event

An occurrence of some activity within a facility. There are two types of events: **duration** and **point**. Duration events have logical beginning and ending points. Point events occur instantaneously. You can define up to 128 events for each facility.

event name

A 1- to 15-character string that names an event.

facility

Software, usually referred to as a layered product, that serves a particular purpose and operates under OpenVMS. See also **application program**.

facility definition

A description of the events and items that Oracle Trace can capture for a particular facility. Each facility for which Oracle Trace can collect data must have a facility definition stored in the Administration database.

facility selection

A description of what to collect during data collection. Facility selections include a list of facilities and their collection classes. One or more data collections can use the same facility selection.

formatted data file

A file that contains data organized for reporting. The formatted data file can contain data from one or more data files.

frequency

How many times an event occurred in the collection interval.

History database

A single Oracle Rdb database that contains all of the informational and error messages associated with data collection. There is one History database per system or cluster. You reference the History database with the logical name EPC$HISTORY_DB.

instrumenting

The act of adding Oracle Trace service routine calls to an application program so that event data can be collected.

interval

A time period when data collection takes place.

item

A numeric or string value that can be collected for an occurrence of an event. Up to 128 items can be captured for each event.

item name

A 1- to 15-character string representing the name of an item.

local collection

A collection that is active or pending either on a standalone system or on one node in a cluster as opposed to data collection scheduled clusterwide.

local node

The system you are currently logged into.

Oracle Trace service routines

See **service routines**.

point event

See **event**.

registrar process

A detached process that handles all communication between the applications instrumented with Oracle Trace routine calls and the Oracle Trace Administration database.

registration identifier

A 1- to 255-character string that the EPC$INIT service routine passes to Oracle Trace. It is useful for distinguishing separate images that use the same facilities. You can use the /REGISTRATION_ID qualifier to the SCHEDULE COLLECTION command to limit data collections to processes with a specific registration ID, EPID, image name, user name or process name.

remote

Pertaining to or originating from another node in the network.

reporting

The process of creating reports based on a formatted file or database. Oracle Trace can create reports based on data stored in an Oracle Rdb database.

resource utilization items

A set of standard items that Oracle Trace collects for all facilities. The items are referenced by the group name RESOURCE_ITEMS.

service routines

A set of predefined Oracle Trace routines whose calls are instrumented in the source code of an application program so that event data can be collected.

source program

A program that expresses an algorithm in a programming language such as FORTRAN, COBOL, or Pascal. After a source program is instrumented with Oracle Trace service routine calls, it is referred to as a facility.

start event

The beginning of a duration event. See **event**.

Index

P

R

S

T

V