Oracle Trace™

Reporter User's Guide

Version 2.2

Part No. A38161-1

ORACLE®

Oracle Trace Reporter User's Guide

Version 2.2

Part No. A38161-1

Contents

Send Us Your Comments ix

Preface xi

1 Overview

2 Merging and Formatting Data Collection Files

2.1 Merging and Formatting Collected Data into an Oracle Rdb Database 2–1
2.2 How to Optimize Formatting for Oracle Rdb Databases 2–3
2.3 Merging and Formatting Collected Data into a VAX RMS File 2–4

3 Producing Standard Oracle Trace Reports

3.1 Report Index Pages 3–2
3.2 Using REPORT Qualifiers to Change Report Content and Organization 3–2
3.2.1 Joining Items Between Events and Facilities 3–3
3.2.2 Reporting on Individual Event Occurrences 3–4
3.3 Summary Report 3–4
3.3.1 Specifying the Kinds of Statistics to Report 3–4
3.4 How Oracle Trace Reports on Duration Events 3–7
3.4.1 Limiting Items in a Report 3–9
3.4.2 Connecting Events by Shared Item Values Using /JOIN 3–10
3.4.3 Displaying All Occurrences of an Event Using /GROUP_BY 3–16
3.5 Detail Report 3–21
3.6 Frequency Report 3–24
3.7 How Oracle Trace Wraps Text in Reports 3–26

4 Producing Customized Oracle Trace Reports

4.1 Keywords to the /OPTIONS Qualifier 4–2
4.2 Using the /OPTIONS Qualifier 4–2
4.3 Example of a Customized Report 4–4
4.4 Connecting Events Using the JOIN Keyword 4–13

5 Optimizing the Report Process

5.1 How to Optimize Reporting from Large Formatted Databases 5–1
5.2 Optimizing Reports That Use the /JOIN Qualifier 5–2
5.3 Defining an Index 5–3

6 Troubleshooting Oracle Trace Reports

6.1 Field Width Too Small for Value 6–3
6.2 Problems with Reports Using JOIN 6–4

7 Oracle Trace Formatting and Reporting Commands

@ (Execute Procedure) 7–3
FORMAT 7–4
HELP 7–11
REPORT 7–12
REPORT Options—EVENT 7–20
REPORT Options—ITEM 7–24
REPORT Options—JOIN 7–25
REPORT Options—RESTRICTION 7–26

A Oracle Rdb Database Format

A.1 The Data Types A–2
A.2 String Data Segmentation A–3
A.3 Relations for the Control Information A–5
A.3.1 The EPC$IDENT Relation A–5
A.3.2 The EPC$SELECTION Relation A–5
A.3.3 The EPC$COLLECTION Relation A–6
A.3.4 The EPC$DCF Relation A–7
A.3.5 The EPC$REG Relation A–8
A.3.6 The EPC$FACILITY Relation A–8
A.3.7 The EPC$EVENT Relation A–9
A.3.8 The EPC$ITEM Relation A–10

A.3.9 The EPC$EVENT_ITEM Record A–12
A.3.10 The EPC$PROCESS Relation A–13
A.3.11 The EPC$IMAGE Relation A–15
A.3.12 The EPC$DCF_IMAGE Relation A–16
A.3.13 The EPC$FACILITY_IMAGE Relation A–16
A.3.14 The EPC$RELATIONSHIP_TABLE_NAMES Relation A–17
A.4 Relations for the Collected Data A–18
A.5 The Entity Relationship Diagram for the Database A–19
A.6 The Predefined Views A–20
A.6.1 Relating Collection Information with Event Data A–21
A.6.2 Relating Selection and Collection Information with Event Data A–21
A.6.3 Relating Image Information with Event Data A–21
A.6.4 Relating Process and Image Information with Event Data A–21
A.6.5 Limitations A–22

B VAX RMS File Format

B.1 Data Types B–1
B.1.1 Date B–1
B.1.2 Fixed ASCIC/ASCIW String B–2
B.2 Records Organization B–2
B.2.1 Control Records B–3
B.2.1.1 FMTDICT Record B–3
B.2.1.2 COLLECTION Record B–4
B.2.1.3 FACILITY Record B–4
B.2.1.4 DCF Record B–5
B.2.1.5 IMAGE Record B–5
B.2.1.6 FACILITY-REGISTRATION Record B–7
B.2.1.7 EVENT Record B–8
B.2.1.8 Item Descriptor B–9
B.3 Data Collection Records B–10

Index

Examples

3–1	Example Report Index	3–2
3–2	Summary Report Segment Showing a Duration Event	3–6
3–3	Summary Report Segment Showing a Point Event	3–8
3–4	Summary Report Segment Showing Only the Elapsed Item	3–9
3–5	Summary Report Showing the Balance Event and Its Underlying Oracle Rdb Transactions	3–11
3–6	Report with Two ATM Events Connected	3–14
3–7	Summary Report Showing Events Grouped According to Account Number	3–17
3–8	Detail Report Segment Showing Balance Event Items	3–22
3–9	Sample Frequency Report Based on ATM Data	3–25
3–10	Wrapping of a Groupable Item	3–26
3–11	Wrapping Items	3–27
3–12	Wrapping of Header Text	3–28
3–13	Wrapping of Item Text	3–28
4–1	OPTIONS.OPT Example Options File	4–3
4–2	Command Segment Specifying the First Subreport	4–4
4–3	First Event-Specific Subreport Produced by the Command in Example 4–2	4–5
4–4	Command Segment Specifying the Second Subreport	4–6
4–5	Second Subreport Example Produced by the Command in Example 4–4	4–7
4–6	Command Specifying the Third Subreport	4–8
4–7	Third Subreport Example Produced by the Command in Example 4–6	4–9
4–8	Fourth Subreport Command	4–11
4–9	Fourth Subreport Example Produced by the Command in Example 4–8	4–12
4–10	Example Command Connecting Two Events	4–13
4–11	Example Report with Two Connected Events	4–14
5–1	Oracle Rdb Debug Display	5–4
6–1	Example of Specifying Incorrect Items for a Join	6–5
6–2	Example of a Failed Join Report	6–5

Figures

A–1 The Entity Relationship Diagram for the Database A–20

Tables

1–1 Commands for Generating Reports 1–2
2–1 Format Optimization Parameters 2–3
3–1 Oracle Trace Reports 3–1
3–2 How Summary Reports Treat Duration Events 3–7
3–3 Text Wrapping in Oracle Trace Reports 3–26
5–1 How to Identify and Use an Index 5–2
6–1 Common Reporting Errors 6–1
6–2 The Format of Summary Report Statistics 6–4
7–1 Oracle Trace Commands 7–2
7–2 Oracle Rdb Optimization Parameters 7–7
7–3 Oracle Trace Reports 7–18
7–4 Interaction Between REPORT Qualifiers and Their Corresponding /OPTIONS Keyword Qualifiers 7–20
A–1 Fields Not Identically Defined A–1
A–2 Oracle Rdb Representations of Oracle Trace-Supported Data Types A–2
A–3 The Point Event Relation EPC$1_40_REQUEST A–4
A–4 A Relation for Segmented Strings A–4
A–5 EPC$IDENT Relation A–5
A–6 EPC$SELECTION Relation A–6
A–7 EPC$COLLECTION Relation A–6
A–8 EPC$DCF Relation A–7
A–9 EPC$REG Relation A–8
A–10 EPC$FACILITY Relation A–8
A–11 EPC$EVENT Relation A–9
A–12 EPC$ITEM Relation A–11
A–13 EPC$EVENT_ITEM Relation A–12
A–14 EPC$PROCESS Relation A–14
A–15 EPC$IMAGE Relation A–15
A–16 EPC$DCF_IMAGE Relation A–16
A–17 EPC$FACILITY_IMAGE Relation A–16
A–18 Relationship Table A–18

A–19	Event Relation Fields	A–19
B–1	Record Type Literal	B–2
B–2	FMTDICT Record	B–3
B–3	COLLECTION Record	B–4
B–4	FACILITY Record	B–4
B–5	DCF Record	B–5
B–6	IMAGE Record	B–6
B–7	FACILITY-REGISTRATION Record	B–7
B–8	Event Record Description	B–8
B–9	Item Descriptor	B–9
B–10	Data Collection Record	B–10

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 describes how to use Oracle Trace for OpenVMS Version 2.2 reporting capabilities. The documentation refers to Oracle Trace for OpenVMS by its abbreviated name: Oracle Trace. Oracle Trace is a layered product that gathers and reports event-based data from any combination of OpenVMS layered products and application programs containing Oracle Trace service routine calls. The Oracle Trace documentation refers to application programs that contain Oracle Trace service routines as **facilities**. The following Oracle products are facilities:

- Oracle CODASYL DBMS
- Oracle RALLY
- Oracle Rdb

Oracle Trace provides Oracle Expert with data that it uses for optimizing existing Oracle Rdb databases.

Additionally, the following Digital Equipment Corporation products are also facilities:

- ACMS
- ALL–IN–1
- DECforms

You can collect event-based data from any products that contain Oracle Trace service routine call. You can also add Oracle Trace service routines to your own applications to collect data from them. The process of adding Oracle Trace service routine calls to an application is called **instrumenting** an application. The products that are instrumented for Oracle Trace provide documentation that describes details of their instrumentation.

Oracle Trace software operates with minimal performance impact on the system. It can run with both the development and production versions of your application to give you information about the behavior of your application.

Intended Audience

This manual is intended for application programmers, software performance analysts, and database administrators who want to generate reports of event-based data.

Operating System Information

Oracle Trace Version 2.2 requires OpenVMS VAX Version 5.4-3 higher or OpenVMS Alpha Version 1.5 or higher.

Structure

This manual has the following structure:

Chapter 1	Provides an overview of the Oracle Trace reporting software.
Chapter 2	Describes how to merge and format event-based data from one or more data collection files.
Chapter 3	Describes how to produce standard Oracle Trace reports.
Chapter 4	Describes how to produce customized Oracle Trace reports.
Chapter 5	Describes how you can optimize the reporting process. Oracle Corporation recommends that you read this chapter before you use the JOIN qualifier or keyword.
Chapter 6	Describes the Oracle Trace Format and Report commands.
Chapter 7	Describes common reporting problems, their symptoms, and how to correct them.
Appendix A	Describes the formatted database produced by the Oracle Trace formatter.
Appendix B	Describes the layout of the VAX RMS file produced by the Oracle Trace formatter.

Related Documents

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 th
- *Oracle Trace Collector User's Guide* Version 2.2—Describes how to collect event-based data and instrument applications for collection.
- *Oracle Trace Getting Started* Version 2.2—Provides an introduction to the Oracle Trace software and instructions for scheduling a collection and generating a report.

Conventions

The special symbols used in this book are:

Symbol	Meaning
Ctrl/x	This symbol tells you to press the Ctrl (control) key and hold it down while pressing the specified letter key.
Bold	Bold lettering in text indicates the definition of a new term.
[ ]	Brackets indicate optional elements.
. . .	Horizontal ellipsis indicates that you can enter additional parameters, values, or information.
. . .	Vertical ellipsis in an example means that information not directly related to the example has been omitted.
$	The dollar sign is used to indicate the DCL prompt. This prompt may be different on your system.

References to Products

In this manual Oracle Trace for OpenVMS is more simply referred to as Oracle Trace.

The naming conventions for other Oracle products discussed in this guide are as follows:

- Oracle Rdb refers to all products previously known as: DEC Rdb for OpenVMS VAX, DEC Rdb for OpenVMS AXP, and VAX Rdb/VMS.
- Oracle CDD/Repository for OpenVMS VAX and Oracle CDD/Repository for OpenVMS Alpha software are referred to as Oracle CDD/Repository or the dictionary. (Previous to Version 5.0, Oracle CDD/Repository was called VAX CDD/Plus.)
- Oracle CODASYL DBMS refers to all products previously known as: DEC DBMS for OpenVMS VAX, DEC DBMS for OpenVMS AXP, and VAX DBMS.

The naming conventions for Digital Equipment Corporation products discussed in this guide are as follows:

- DEC DATATRIEVE is referred to as DATATRIEVE.
- DEC ACMS is referred to as ACMS.
- VAX Language-Sensitive Editor software is referred to as LSE.
- Various Digital Equipment Corporation programming language compilers are referred to by their common names, without the VAX or DEC prefix.

New Features

Oracle Trace Version 2.2 includes a BLR to SQL converter and a new method of analyzing data using Microsoft Excel. These features are described in the following sections.

BLR to SQL Converter

This release of Oracle Trace contains a tool that helps in the diagnosis of Oracle Rdb data. This tool, the BLR to SQL converter, translates the BLR field of the REQUEST_BLR event into an SQL query. The converter works on a formatted Oracle Trace database and requires the Oracle Rdb ALL or RDBEXPERT class of data.

The BLR to SQL converter takes the following steps:

1. Attaches to the Oracle Trace formatted database and creates a new table, EPC$SQL_QUERIES.
2. Processes the BLR strings in the EPC$1_221_REQUEST_BLR table.
3. Converts the BLR strings to SQL queries.

4. Stores the SQL queries in EPC$SQL_QUERIES.
5. Collapses similar requests to the same SQL.
6. Generates a unique SQL_ID.

   The SQL_ID is a new field that is added to both the EPC$1_221_REQUEST_BLR table and the EPC$1_221_REQUEST_ACTUAL table.

After running the converter, you can perform queries against the formatted database to associate critical requests with their respective SQL statement. An example of an SQL join that links new information across multiple tables in the formatted database is provided in EPC$EXAMPLES:SAMPLE_SQL_JOIN.SQL.

To help you use the converter, a command procedure, EPC_BLR_TO_SQL_CONVERTER.COM, is available in EPC$EXAMPLES. The command procedure is thoroughly commented with explanations of all options, allowing you to tailor it to meet your own needs.

Analyzing Oracle Trace Data Using Microsoft Excel

You can access an Oracle Trace formatted database from Microsoft Excel or other PC analysis and reporting tools.

If you have Microsoft Excel Version 5.0, you can use ODBC 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 (located in the EPC$EXAMPLES directory) show you how to access the data in an Oracle Trace formatted database for the purpose of analysis using Microsoft Excel.

EPC$ACTUALS_VIEW.COM
EPC$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 on the COLLECT FORMAT command, if you want 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.

1
Overview

The Oracle Trace Reporter software produces either a formatted Oracle Rdb database or a VAX RMS file from one or more Oracle Trace data collection files. It also produces tabular reports from the formatted Oracle Rdb database. These reports display event-based data and resource use information.

The Oracle Trace Reporter software is very flexible and provides many reporting capabilities:

- You can produce three kinds of standard reports:
  - Summary Report — Provides statistical evaluations of resource consumption by event. This information is useful for performance evaluation and capacity planning.
  - Detail Report—Provides values of items collected for each event. This report is useful for debugging an application that you are instrumenting.
  - Frequency Report—Provides a count of event occurrences for each interval within a collection. This report is useful for tracing the execution of an instrumented application.
- You can produce customized Oracle Trace reports that contain unique subreports with characteristics tailored to the event they are reporting.
- You can create your own reports based on the formatted files or database that Oracle Trace produces.

Producing a report is a two step process:

1. Format one or more data collection files using the FORMAT command as described in Chapter 2.
2. Produce standard reports from the formatted Oracle Rdb database using the REPORT command as described in Chapter 3.

As you become more acquainted with Oracle Trace reporting capabilities, you can produce customized reports as described in Chapter 4, or produce your own reports based on either the formatted Oracle Rdb database or VAX RMS file that the Oracle Trace formatter produces. Appendix A describes the format of the Oracle Rdb database, and Appendix B describes the layout of the VAX RMS file.

Table 1–1 summarizes the Oracle Trace commands associated with reporting.

Table 1–1 Commands for Generating Reports

Command	Description
@(Execute Procedure)	Executes an indirect command file that contains Oracle Trace commands.
FORMAT	Formats one or more Oracle Trace data files into a formatted database or data file.
HELP	Displays information about Oracle Trace and Oracle Trace commands.
REPORT	Generates a report based on formatted data from one or more collections.

Chapter 7 describes these commands in detail.

Note

Instrumented products provide documentation about the events and items they have instrumented. Refer to the documentation associated with the facilities on which you are reporting for this information. An understanding of how facilities are instrumented will help you to produce more useful reports.

2

Merging and Formatting Data Collection Files

The FORMAT command produces formatted output from one or more Oracle Trace data collection files. You can use the /TYPE qualifier of the FORMAT command to specify two kinds of output:

- An Oracle Rdb database from which you can produce both standard and customized Oracle Trace reports and your own reports using a programming or fourth-generation language. Section 2.1 describes how to produce an Oracle Rdb database, and Section 2.2 tells how you can optimize this process.
- A VAX RMS file from which you can produce your own reports using a programming or fourth-generation language. Section 2.3 describes how to produce a VAX RMS file.

2.1 Merging and Formatting Collected Data into an Oracle Rdb Database

Formatting collected data into an Oracle Rdb database allows you to generate both standard and customized Oracle Trace reports and provide input to other products such as Oracle Expert for Rdb. You can also generate reports from the Oracle Rdb database using a fourth-generation language or a programming language that contains embedded calls to either Oracle Rdb or VAX RMS. Appendix A describes the format of an Oracle Rdb database.

If you format data into an Oracle Rdb database, you can optionally store the record definitions in CDD/Repository using the /CDDPLUS_DEFINITIONS qualifier. Of course, this feature is only available if CDD/Repository is currently installed on your system. If you format the data collection files without the /CDDPLUS_DEFINITIONS qualifier and later decide that you want to use CDD/Repository, you can use the INTEGRATE command of RDO or SQL. This section provides example commands for formatting collected data into an Oracle Rdb database.

You can merge multiple data collection files into a single formatted database. The following example formats the data from two files (COLL1_FILE1.DAT and COLL1_FILE2.DAT) into an Oracle Rdb database called COLL_DB:

```
$ COLLECT FORMAT/TYPE=RDBVMS COLL1_FILE1.DAT,COLL1_FILE2.DAT USER1:COLL_DB
```

If you wish to merge two more data collection files from another collection (using the same facility selection) into the Oracle Rdb database created by the previous command, use the following command:

```
$ COLLECT FORMAT/MERGE/TYPE=RDBVMS COLL2_FILE1.DAT,COLL2_FILE2.DAT -
_$ USER1:COLL_DB
```

If you are formatting one large collection file, you can speed up the formatting operation by specifying the /NOJOURNAL qualifier on the FORMAT command. Although specifying /NOJOURNAL speeds the formatting process, it also eliminates the possibility of recovering the database if a formatting error occurs. If an error occurs during formatting, the database becomes corrupt from the point where the error occurred. If this happens when you are formatting one file, you can repeat the formatting process. However, if a failure occurs when you are merging one or more collection files with a database, the database could become corrupt, and you could lose data. Therefore, Oracle Corporation recommends that you back up the database before you merge collection files with it by using the /NOJOURNAL qualifier. Here is an example:

```
$ RMU/BACKUP USER1:COLL_DB USER1:COLL_DB.RDF -
$ COLLECT FORMAT/NOJOURNAL -
$ /MERGE/TYPE=RDBVMS COLL2_FILE1.DAT,COLL2_FILE2.DAT -
_$ USER1:COLL_DB
```

If an error occurred during the previous formatting operation, you could use a command similar to the following to restore the formatted database before entering the FORMAT command again:

```
$ RMU RESTORE NEW_VERSION USER1:COLL_DB.RDF
```

When formatting several data collection files into a single formatted database, you should format the largest file first and then merge the remaining files into the newly formatted database.

2.2 How to Optimize Formatting for Oracle Rdb Databases

You can improve response time for formatting data collection files into an Oracle Rdb database by specifying a set of optimization parameters on the FORMAT command. The parameters are part of the /RDBVMS_OPTIMIZATION qualifier, which has the following format:

/RDBVMS_OPTIMIZATION = (parameter=value[, . . .])

Table 2–1 shows the optimization parameters and their defaults.

Table 2–1 Format Optimization Parameters

Optimization Parameter	Default Value
ALLOCATION	2000 database pages
BUFFER_SIZE	30 blocks
[NO]JOURNAL	JOURNAL
MIN_EXTENT	500 pages
NUM_BUFFERS	30 buffers
[NO]STRING_OPTIMIZATION	STRING_OPTIMIZATION
[NO]VIEWS	VIEWS

Use the following formula to determine the recommended value for the ALLOCATION parameter:

$$number\text{-}of\text{-}pages = (4000 + size\text{-}of\text{-}DCF)/2$$

Where *size-of-DCF* is the size in blocks of all data collection files.

The default allocation is 2000 pages.

The following example uses the optimization parameter to format a large data collection file containing 16,000 blocks into a formatted Oracle Rdb database:

```
$ COLLECT FORMAT VERY_BIG_FILE.DAT BIG_DATABASE -
_$ /RDBVMS_OPTIMIZATION=(NOJOURNAL, ALLOCATION=10000 -
_$ NOVIEWS)
```

See the description of the FORMAT command in Chapter 7 for a description of each optimization parameter.

2.3 Merging and Formatting Collected Data into a VAX RMS File

If you format collected data into a VAX RMS file, you must generate reports using a fourth-generation language or a programming language that contains embedded calls to VAX RMS. Appendix B describes the layout of a VAX RMS file.

Note

You cannot use Oracle Trace to produce reports from VAX RMS files.

This section provides example commands for formatting collected data into VAX RMS files.

To merge two data collection files from a single collection into a new VAX RMS file called TWO_DAYS.DAT, use the following command:

```
$ COLLECT FORMAT/TYPE=RMS COLL1_FILE1,COLL1_FILE2 USER1:TWO_DAYS.DAT
```

If you want to merge two more data collection files from another collection (using the same facility selection) into the same VAX RMS file that was just created, issue the following command:

```
$ COLLECT FORMAT/MERGE/TYPE=RMS COLL2_FILE1,COLL2_FILE2 -
_$ USER1:COLL_FILE
```

3

Producing Standard Oracle Trace Reports

You can use the Oracle Trace REPORT command to produce tabular reports based on data in an Oracle Rdb formatted database produced by the FORMAT command. Table 3–1 describes the three kinds of standard reports that you can specify with the /TYPE qualifier of the REPORT command.

Table 3–1 Oracle Trace Reports

Report Name	Purpose	Description	Associated Qualifiers	See Section
Summary (default)	For evaluating event resource use for performance and capacity planning purposes.	Summarizes statistics about the collected data.	/ITEMS[1], /STATISTICS[2], /FACILITIES, /EVENTS, /GROUP_BY, or /OPTIONS	3.3
Detail	For debugging instrumented applications or application debugging.	Contains values of the items collected for each event.	/FACILITIES, /EVENTS, and /ITEMS[1] or /OPTIONS as described in Chapter 4.	3.5
Frequency	For tracing the execution of an instrumented application.	Summarizes event occurrences based on a selected interval of seconds, minutes, or hours.	/INTERVAL[3], /FACILITIES, /EVENTS, /GROUP_BY, or /OPTIONS	3.6

[1]The /ITEMS qualifier is valid only for the Detail and Summary reports.

[2]The /STATISTICS qualifier is valid only for Summary reports.

[3]The /INTERVAL qualifier is valid only for Frequency reports.

All of the Oracle Trace reports display information for events based on the criteria you specify in the REPORT command. Except for the qualifiers noted in Table 3–1, you can use all of the REPORT command qualifiers for each kind of report. You can use the /TYPE qualifier with the SUMMARY, DETAIL, or FREQUENCY keywords to specify the kind of report you want. If you do not

specify a report, Oracle Trace produces a Summary report by default. See Chapter 7 for a description of the REPORT command and its qualifiers.

3.1 Report Index Pages

All reports contain an index page that lists the facilities, the events in each facility, and the report page on which the events are reported. Example 3–1 is an example report index.

Example 3–1 Example Report Index

```
Index                                Page 7
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2
Report Index

Facility Name              Event Name                    Join item  Page

ATM_SAMPLE                 Balance                          2          1
RDBVMS                     Transaction                      2          3
```

3.2 Using REPORT Qualifiers to Change Report Content and Organization

You can affect both the organization and the content of reports by the way you use the /FACILITY, /EVENTS, or /ITEMS qualifiers. For example, if you do not specify values for the /FACILITY, /EVENTS, or /ITEMS qualifiers, Oracle Trace reports on all of the facilities, events, and items in the formatted database. When you specify a value for the /FACILITY, /EVENTS, or /ITEMS qualifier, Oracle Trace limits the report to the facilities, events, and items that you specify.

Oracle Trace groups information hierarchically with facilities at the highest level, events at the next level, and items at the lowest level. It arranges information within each level alphabetically. (The one exception is the Elapsed item which always appears first.) Therefore, by default, the ACMS facility would precede the Oracle CODASYL DBMS facility in reports, and within a facility, the balance event would always precede the Transaction event. Within an event, Buffered IO would always precede CPU, and so on.

In addition to limiting the report, you can also use these qualifiers to override the default alphabetical arrangement Oracle Trace imposes on report information. For example, if you were reporting on the Oracle Rdb and the ACMS facilities and want the information for the Oracle Rdb facility to precede

the information for the ACMS facility in the report, you specify the facilities in the order you want them to appear, as follows:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/FACILITY=(RDBVMS,ACMS) -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/OUTPUT=SAMPLE_SUMMARY_REPORT.RPT
```

3.2.1 Joining Items Between Events and Facilities

The /JOIN qualifier connects events and groups them within a report based on a shared data item. This capability allows you to track resource use across events and facilities. For example, you can now relate the ACMS Procedure Call event with its related Oracle Rdb events for a greater understanding of the total resources the ACMS Procedure Call uses.

In addition to limiting the report to specific events, the /EVENT qualifier also affects the arrangement of joined items in a report when you use it in conjunction with the /JOIN qualifier. The order in which you specify events determines the report hierarchy. Oracle Trace displays the first event and its related items first, with subsequent events and their items under it. This arrangement allows you to display related events and items logically.

Joined items must be of the same data type and must share a value for an item. If you want to use the /JOIN qualifier successfully, you must have a general understanding of how events come to share a value for an item, and a knowledge of the specific events that share item values in the facilities on which you are reporting.

If you are reporting on Oracle-supplied facilities such as Oracle Rdb, consult the documentation accompanying these facilities to learn which events share item values or examine the facility definitions for these facilities. Be sure that the events that you want to join share an item value.

If you are reporting on an application that you have instrumented with Oracle Trace routines, you might be able to determine which events share which item values based on how you related these events in your application. Some relations are defined using the RELATE keyword of the /OPTIONS qualifier of the CREATE DEFINITION command. You can also examine the facility definitions for these facilities to determine if the events that you want to join share the same item value.

3.2.2 Reporting on Individual Event Occurrences

You can use the /GROUP_BY qualifier to report on individual event occurrences. The /GROUP_BY qualifier groups event occurrences based on an item that is unique for each event occurrence. For example, in Oracle Rdb the CLIENT_PC item contains the value of the program counter each time the event executes, and in the ATM_SAMPLE, the ACCOUNT_ID contains the number of the account to which an event pertains. Specifying these items with the /GROUP_BY qualifier displays each occurrence of the events associated with these items.

For Summary reports, Oracle Trace does not display string or level items such as the ACCOUNT_ID in the ATM_SAMPLE because the values of these items are descriptive rather than numerical. If you want to see string items in Summary reports, you must use the /GROUP_BY qualifier and specify the string or level item.

3.3 Summary Report

The Summary report is the default report type. You can use the information in the Summary report for performance evaluation or capacity planning. The Summary report computes statistics based on the data collected for all occurrences of an event and summarizes this data for each event. The Summary report helps you to evaluate the resources each event uses.

By default, a Summary report displays all statistics for all items associated with every event from every facility in the formatted Oracle Rdb database. You can use the /STATISTICS, /FACILITY, /EVENT, and /ITEM qualifiers to limit the report to specific statistics, facilities, events, and items.

3.3.1 Specifying the Kinds of Statistics to Report

The Summary report organizes information by event and displays statistics for all of the items associated with each event.

You can use the following /STATISTICS qualifier keywords to specify the kinds of statistics that you want to report:

ALL	MEAN	TOTAL
MAXIMUM	STANDARD_DEVIATION	95TH_PERCENTILE[1]
MINIMUM	COUNT (number of occurrences)	

[1]The 95th percentile is based on the standard deviation. It assumes a relatively normal distribution with a sample size, n, greater than 1000. Based on sample sizes greater than 1000, a confidence level of 95% can be computed as 1.96 units away from the standard deviation.

If you do not use the /STATISTICS qualifier Oracle Trace reports COUNT as the default. The sample standard deviation is computed by the following formula, where n is the sample size and X_i is the value of each sample element:

$$Sample_i = \sqrt{\frac{n\sum_{i=1}^{n} X_i^2 - (\sum_{i=1}^{n} X_i)^2}{n(n-1)}}$$

Oracle Trace stores the subvalues it uses to calculate the sample standard deviation as D-floating data types. These types have a range of 0.29 * 10E-38 to 0.29 * 10E38 with 16 decimal places of precision. If the value of the squared elements summed requires more than 16 decimal places, a roundoff error may occur.

Note

Oracle Trace stores the value of facility-specific Oracle Rdb code_ quadword items as quadwords. Displaying these items as a decimal type can cause a roundoff error. Oracle Trace internally stores quadwords displayed as decimals as D-floating point. The precision is only 16 decimal-digits with a range of .29*10E-38 to 1.7*10E38. Oracle Corporation recommends that you display these values in hexadecimal. You can do this by modifying the facility definition to add /RADIX=HEXADECIMAL to the ITEM option on the CREATE DEFINITION command.

Use a command similar to the following to create a Summary report with all statistics:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/OUTPUT=SAMPLE_SUMMARY_REPORT.RPT
```

Example 3–2 shows part of a Summary report for a duration event.

Example 3–2 Summary Report Segment Showing a Duration Event

```
19-AUG-1994 19:25                Summary Report                        Page 1
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                        14  T PRIO

Minimum       1.10           13            2           2       9            0
Maximum     144.12           30           78           2       9           71
Mean         10.20        14.00         7.41        2.00    9.00         4.17
Std Dev      34.51         4.12        18.20        0.00    0.00        17.22
95 Prct      77.86        22.08        43.08        2.00    9.00        37.92
Total       173.47          238          126          34     153           71
Count           17

           PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                             IOs         SIZE

Minimum             0            0        11219          674         2203
Maximum           816           35        11547          789         2329
Mean            48.11         2.05     11527.70       781.17      2316.29
Std Dev        197.87         8.48        79.55        27.69        33.24
95 Prct        435.96        18.69     11683.62       835.45      2381.45
Total             818           35       195971        13280        39377
Count              17

             WORKING
             SET SIZ

Minimum         4005
Maximum         4517
Mean         4486.88
Std Dev       124.17
95 Prct      4730.27
Total          76277
Count             17

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

```
   .
   .
   .
```

The facility definition associated with the collection defines the characteristics of item headers, such as CURRENT_PRIO, DIRECT_IO, and PAGE_FAULTS. You can override the default header text and width using the ITEMS keyword to the /OPTIONS qualifier described in Chapter 4.

3.4 How Oracle Trace Reports on Duration Events

When you collect standard resource items, Oracle Corporation recommends collecting the same items on both the EPC$START_EVENT(W) and EPC$END_EVENT(W) routine calls. However, for facility-specific items, each instrumented application determines what gets collected on each EPC$START_EVENT(W) and EPC$END_EVENT(W) routine call. Therefore, Oracle Trace does not always collect items the same way for duration events. Table 3–2 describes how Summary reports handle information associated with duration events.

Table 3–2 How Summary Reports Treat Duration Events

For Duration Events with . . .	the Summary Report displays . . .
unmatched EPC$START_EVENT(W) and EPC$END_EVENT(W) routine calls	a diagnostic message following the event indicating how many unmatched event records were encountered. If grand totals are printed for the report, it includes the total number of unmatched event records.
items specified for collection on both the EPC$START_EVENT(W) and EPC$END_EVENT routine calls	only the values associated with the item at the EPC$END_EVENT(W) call.
items specified for collection on either an EPC$START_EVENT(W) or an EPC$END_EVENT(W) call	the values associated with the corresponding EPC$START_EVENT(W) or EPC$END_EVENT(W) call.

Example 3–3 shows part of a Summary report for a point event.

Example 3–3 Summary Report Segment Showing a Point Event

```
   .
   .
   .
19-AUG-1994 19:25                Summary Report                        Page 3
Selection: SAMPLE_SELECTION                                    Oracle Trace V2.2

Event:    Errors          In Facility:  ATM_SAMPLE      Version:  V2.2

         Timestamp                BUFFERED IO     CPU TIME  CURREN   DIRECT IO
                                                             T PRIO

Minimum  22-JUL-1994 17:45:54.96        42831        27486       7        17601
Maximum  22-JUL-1994 17:55:59.41        55664        29961       8        18839
Mean                                 49111.50     28702.33    7.77     18215.72
Std Dev                               4164.97       808.23    0.42       398.00
95 Prct                              57274.85     30286.48    8.61     18995.81
Total                                  884007       516642     140       327883
Count    18

          PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                              IOs         SIZE

Minimum        68374         3097        11219          680         2203
Maximum        68674         3097        11547          789         2329
Mean         68645.88      3097.00     11528.55       780.83      2316.27
Std Dev         72.01         0.00        77.26        25.33        32.17
95 Prct      68787.04      3097.00     11679.98       830.48      2379.33
Total         1235626         55746       207514        14055        41693
Count              18

            WORKING
            SET SIZ

Minimum         4005
Maximum         4517
Mean         4488.55
Std Dev       120.67
95 Prct      4725.08
Total          80794
Count             18
%EPC-I-RPQU_BAD_95, 95 Prct for events with counts under 1000 are less precise
   .
   .
   .
```

3.4.1 Limiting Items in a Report

You use the /ITEMS qualifier to limit a report to specific items for each event. You can also produce a Summary report that displays only the Elapsed item by specifying /NOITEMS as shown in the following example:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/FACILITY=RDBVMS,ACMS -
 _$/TYPE=SUMMARY -
 _$/NOITEMS -
 _$/STATISTICS=ALL -
 _$/OUTPUT=SAMPLE_SUMMARY_REPORT.RPT
```

Example 3–4 is an example report that displays only the Elapsed item for each event.

Example 3–4 Summary Report Segment Showing Only the Elapsed Item

```
20-AUG-1994 16:53               Summary Report                        Page 1
Selection: SAMPLE_SELECTION                                 Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

           Elapsed

Minimum       1.10
Maximum     144.12
Mean         10.20
Std Dev      34.51
95 Prct      77.86
Total       173.47
Count           17

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

20-AUG-1994 16:53               Summary Report                        Page 2
Selection: SAMPLE_SELECTION                                 Oracle Trace V2.2

Event:  Deposits        In Facility:  ATM_SAMPLE      Version:  V2.2

           Elapsed

Minimum       0.38
Maximum       4.10
Mean          0.78
Std Dev       0.60
95 Prct       1.97
Total        96.61
Count          123

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

(continued on next page)

Example 3–4 (Cont.) Summary Report Segment Showing Only the Elapsed Item

```
20-AUG-1994 16:53                Summary Report                          Page 3
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Errors          In Facility:  ATM_SAMPLE      Version:  V2.2

         Timestamp

Minimum  22-JUL-1994 17:45:54.96
Maximum  22-JUL-1994 17:55:59.41
Mean
Std Dev
95 Prct
Total
Count    18

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

3.4.2 Connecting Events by Shared Item Values Using /JOIN

You can use the /JOIN qualifier to connect events based on the value of an item they share. For example, the CROSS_FAC_14 item is associated with events in both the ATM_SAMPLE and the Oracle Rdb facility.

Note

Oracle Corporation recommends that you read the optimizing recommendations in Chapter 5 when you use the /JOIN qualifier.

You can use a command similar to the following to display the ATM events associated with the CROSS_FAC_14 item and their underlying Oracle Rdb transactions:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/FACILITY=(ATM_SAMPLE,RDBVMS) -
 _$/EVENTS=(BALANCE_EVENT,TRANSACTION) -
 _$/JOIN=(BALANCE_EVENT.CROSS_FAC_14=TRANSACTION.CROSS_FAC_14) -
 _$/OUTPUT=SAMPLE_SUMMARY_REPORT.RPT
```

This command limits the report to the balance event, as shown in Example 3–5.

Example 3–5 Summary Report Showing the Balance Event and Its Underlying Oracle Rdb Transactions

```
24-SEP-1994 18:56                Summary Report                          Page 1
Selection: SAMPLE_SELECTION                                   Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
CROSS FAC 14 : 2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.24           13            1           2       8            0
Maximum       8.28           13           13           2       9            7
Mean          1.02        13.00         3.53        2.00    8.99         0.03
Std Dev       2.23         0.00         1.07        0.00    0.08         0.41
95 Prct       3.45        13.00         5.64        2.00    9.15         0.84
Total       317.02         4017         1092         618    2779           11
Count          309
   .
   .
   .

               ================= Grand Total ====================

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.24           13            1           2       8            0
Maximum       8.28           13           13           2       9            7
Mean          1.02        13.00         3.53        2.00    8.99         0.03
Std Dev       2.23         0.00         1.07        0.00    0.08         0.41
95 Prct       3.45        13.00         5.64        2.00    9.15         0.84
   .
   .
   .
```

(continued on next page)

Example 3–5 (Cont.) Summary Report Showing the Balance Event and Its Underlying Oracle Rdb Transactions

```
24-SEP-1994 18:56                Summary Report                        Page 3
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

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

Join Level =   1
CROSS FAC 14 : 2

           Elapsed     AIJ File  BUFFERED IO       Buffer    Client PC
                         Writes                  Pool Rds

Minimum       0.00           0            0            0        B8073
Maximum       0.34           0            0           14        B8073
Mean          0.01        0.00         0.00         0.04     B8073.00
Std Dev       0.01        0.00         0.00         0.79         0.00
95 Prct       0.05        0.00         0.00         1.60     B8073.00
Total         3.94           0            0           14      DE20ACF
Count          309
   .
   .
   .
24-SEP-1994 18:56                Summary Report                        Page 4
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2
   .
   .
   .
               ================= Grand Total ====================

           Elapsed     AIJ File  BUFFERED IO       Buffer    Client PC
                         Writes                  Pool Rds

Minimum       0.00           0            0            0        B8073
Maximum       0.34           0            0           14        B8073
Mean          0.01        0.00         0.00         0.04     B8073.00
Std Dev       0.01        0.00         0.00         0.79         0.00
95 Prct       0.05        0.00         0.00         1.60     B8073.00
Total         3.94           0            0           14      DE20ACF
Count          309

   .
   .
   .
```

(continued on next page)

Example 3–5 (Cont.) Summary Report Showing the Balance Event and Its Underlying Oracle Rdb Transactions

```
24-SEP-1994 18:56                    Index                              Page 7
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Report Index

Facility Name               Event Name                       Join item  Page

ATM_SAMPLE                  Balance                          2          1
RDBVMS                      Transaction                      2          3
```

If you want to connect two events in the ATM_SAMPLE application, you use a command similar to the following:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/FACILITY=(ATM_SAMPLE) -
 _$/OUT=JOIN_REPORT.RPT -
 _$/EVENTS=(WITHDRAW_EVENT,BALANCE_EVENT)-
 _$/JOIN=(ATM_SAMPLE.WITHDRAW_EVENT.ACCOUNT_ID= -
 _$ATM_SAMPLE.BALANCE_EVENT.ACCOUNT_ID)
```

Example 3–6 is an example report produced by this command.

Example 3–6 Report with Two ATM Events Connected

```
25-SEP-1994 05:51                Summary Report                        Page 1
Selection: SAMPLE_SELECTION                                   Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 1

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.42           13            4           4       9           4
Maximum       3.52           13            9           4       9           5
Mean          2.28        13.00         6.25        4.00    9.00        4.58
Std Dev       0.82         0.00         1.35        0.00    0.00        0.51
95 Prct       2.89        13.00         8.90        4.00    9.00        5.59
Total        15.41          156           75          48     108          55
Count           12
   .
   .
   .
25-SEP-1994 05:51                Summary Report                        Page 2
Selection: SAMPLE_SELECTION                                   Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 1

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.45           13            3           2       9           0
Maximum       8.28           13           13           2       9           7
Mean          2.17        13.00         4.78        2.00    9.00        0.57
Std Dev       2.20         0.00         2.51        0.00    0.00        1.86
95 Prct       6.49        13.00         9.71        2.00    9.00        4.23
Total        30.46          182           67          28     126           8
Count           14
   .
   .
   .
25-SEP-1994 05:51                Summary Report                        Page 3
Selection: SAMPLE_SELECTION                                   Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 2
```

(continued on next page)

Example 3–6 (Cont.) Report with Two ATM Events Connected

```
           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.38           13            2           4       8           4
Maximum       6.16           15            9           4       9          10
Mean          0.97        13.00         5.18        4.00    8.98        4.65
Std Dev       0.88         0.12         1.35        0.00    0.12        0.76
95 Prct       2.71        13.26         7.84        4.00    9.23        6.15
Total       231.43         3096         1235         952    2138        1107
Count          238
   .
   .
   .
25-SEP-1994 05:51                Summary Report                        Page 4
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 2
           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.24           13            1           2       8           0
Maximum       7.91           13            6           2       9           1
Mean          0.97        13.00         3.47        2.00    8.99        0.01
Std Dev       1.15         0.00         0.92        0.00    0.08        0.10
95 Prct       3.23        13.00         5.28        2.00    9.15        0.20
Total       286.55         3835         1025         590    2653           3
Count          295
   .
   .
   .
25-SEP-1994 05:51                     Index                            Page 5
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Report Index

Facility Name               Event Name                     Join item  Page

ATM_SAMPLE                  Withdrawals                     1          1
ATM_SAMPLE                  Balance                         1          2
ATM_SAMPLE                  Withdrawals                     2          3
ATM_SAMPLE                  Balance                         2          4
```

3.4.3 Displaying All Occurrences of an Event Using /GROUP_BY

Although the Summary report produces one set of statistics that represent all occurrences of an event, you can also display specific occurrences of some events by using the /GROUP_BY qualifier. You can use the /GROUP_BY qualifier for events that have an associated item whose value distinguishes event occurrences. For example, in the ATM_SAMPLE program the ACCOUNT_ID item distinguishes event occurrences. If you want to display statistics relating to each occurrence of the balance event, you could use the /GROUP_BY qualifier on the REPORT command as shown in the following example:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/EVENT=WITHDRAW_EVENT -
 _$/FACILITY=(ATM_SAMPLE) -
 _$/GROUP_BY=ACCOUNT_ID -
 _$/OUTPUT=SAMPLE_SUMMARY_REPORT.RPT
```

This command produces a Summary report that displays statistics for all items associated with each occurrence of the balance event, as shown in Example 3–7. Event occurrences are displayed in ascending order based on the value of the ACCOUNT_ID item. The report prints a subtotal of all occurrences of the event and a grand total for the event name.

Example 3–7 Summary Report Showing Events Grouped According to Account Number

```
24-SEP-1994 19:42                Summary Report                        Page 1
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Account Number : 1

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.42           13            4           4       9           4
Maximum       3.52           13            9           4       9           5
Mean          2.28        13.00         6.25        4.00    9.00        4.58
Std Dev       0.82         0.00         1.35        0.00    0.00        0.51
95 Prct       2.89        13.00         8.90        4.00    9.00        5.59
Total        15.41          156           75          48     108          55
Count           12

           PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                             IOs         SIZE

Minimum             0            0        10853         668         2174
Maximum             1            0        10857         675         2245
Mean             0.16         0.00     10853.66      668.91      2230.66
Std Dev          0.38         0.00         1.55        2.23        18.08
95 Prct          0.92         0.00     10856.71      673.29      2266.10
Total               2            0       130244        8027        26768
Count              12

            WORKING
            SET SIZ

Minimum         4005
Maximum         4005
Mean         4005.00
Std Dev         0.00
95 Prct      4005.00
Total          48060
Count             12

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

(continued on next page)

Example 3–7 (Cont.) Summary Report Showing Events Grouped According to Account Number

```
Account Number : 2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                        14  T PRIO

Minimum       0.94           13            3           4       9            4
Maximum       2.22           13            8           4       9            6
Mean          1.33        13.00         5.30        4.00    9.00         4.70
Std Dev       0.45         0.00         1.33        0.00    0.00         0.67
95 Prct       2.21        13.00         7.92        4.00    9.00         6.02
Total        13.35          130           53          40      90           47
Count           10
 . . .

24-SEP-1994 19:42                Summary Report                        Page 2

Account Number : 3

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                        14  T PRIO

Minimum       0.46           13            3           4       8            4
Maximum       6.16           15            8           4       9           10
Mean          2.23        13.08         5.82        4.00    8.91         4.86
Std Dev       1.13         0.41         1.19        0.00    0.28         2.28
95 Prct       3.46        13.90         8.16        4.00    9.47         7.39
Total        28.51          301          134          92     205          112
Count           23
 . . .

Account Number : 4

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                        14  T PRIO

Minimum       0.56           13            3           4       9            4
Maximum       2.56           13            9           4       9            6
Mean          1.18        13.00         5.45        4.00    9.00         4.65
Std Dev       0.49         0.00         1.60        0.00    0.00         0.67
95 Prct       2.15        13.00         8.59        4.00    9.00         5.96
Total        23.60          260          109          80     180           93
Count           20
 . . .
```

(continued on next page)

Example 3–7 (Cont.) Summary Report Showing Events Grouped According to Account Number

```
24-SEP-1994 19:42                Summary Report                        Page 4
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

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

Account Number : 5

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.38           13            3           4       9            4
Maximum       4.84           13            7           4       9            6
Mean          0.94        13.00         5.11        4.00    9.00         4.59
Std Dev       1.12         0.00         1.08        0.00    0.00         0.63
95 Prct       3.16        13.00         7.23        4.00    9.00         5.83
Total        25.63          351          138         108     243          124
Count           27
 . . .
Account Number : 6

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.41           13            2           4       8            4
Maximum       2.74           13            7           4       9            6
Mean          0.70        13.00         4.88        4.00    8.97         4.63
Std Dev       0.44         0.00         1.32        0.00    0.16         0.68
95 Prct       1.56        13.00         7.48        4.00    9.29         5.97
Total        25.30          468          176         144     323          167
Count           36
 . . .
Account Number : 7

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.41           13            2           4       9            4
Maximum       3.40           13            8           4       9            7
Mean          0.87        13.00         5.00        4.00    9.00         4.62
Std Dev       0.62         0.00         1.59        0.00    0.00         0.77
95 Prct       2.09        13.00         8.11        4.00    9.00         6.13
Total        30.63          455          175         140     315          162
Count           35
 . . .
```

(continued on next page)

Example 3–7 (Cont.) Summary Report Showing Events Grouped According to Account Number

```
Account Number : 8

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.41           13            2           4       8            4
Maximum       2.64           13            7           4       9            6
Mean          0.77        13.00         4.95        4.00    8.97         4.63
Std Dev       0.54         0.00         2.23        0.00    0.14         0.67
95 Prct       1.84        13.00         7.37        4.00    9.26         5.95
Total        36.43          611          233         188     422          218
Count           47
 . . .
24-SEP-1994 19:42                Summary Report                        Page 7
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

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

Account Number : 9

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.41           13            3           4       9            4
Maximum       5.75           13            8           4       9            7
Mean          1.19        13.00         5.42        4.00    9.00         4.60
Std Dev       1.37         0.00         1.35        0.00    0.00         0.74
95 Prct       3.90        13.00         8.08        4.00    9.00         6.05
Total        47.95          520          217         160     360          184
Count           40
 . . .
                ================= Grand Total ====================

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.38           13            2           4       8            4
Maximum       6.16           15            9           4       9           10
Mean          0.98        13.00         5.24        4.00    8.98         4.64
Std Dev       0.88         0.12         1.37        0.00    0.12         0.75
95 Prct       2.72        13.25         7.93        4.00    9.23         6.13
Total       246.84         3252         1310        1000    2246         1162
```

(continued on next page)

Example 3–7 (Cont.) Summary Report Showing Events Grouped According to Account Number

```
24-SEP-1994 19:42                Summary Report                        Page 8
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Count          250

           PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                             IOs          SIZE

Minimum             0           0        10853          668         2174
Maximum            63           0        10857          675         2245
Mean             0.27        0.00     10856.84       674.50      2241.95
Std Dev          3.98        0.00         0.78         1.44         6.51
95 Prct          8.08        0.00     10858.37       677.34      2254.71
Total              68           0      2714210       168627       560489
Count             250

             WORKING
             SET SIZ

Minimum         4005
Maximum         4005
Mean         4005.00
Std Dev         0.00
95 Prct      4005.00
Total        1001250
Count            250

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

3.5 Detail Report

The Detail report is useful to application developers when they debug their instrumented applications because it shows the actual values of the items for each start, end, or point event record. For example, you might produce a Detail report to investigate the cause of an unusually high maximum value in a Summary report.

For each event that meets the selection criteria specified in the REPORT command, the Detail report displays a time and date stamp of when the event record was recorded and all report items associated with the event. For duration events, the report displays the values of the items collected on both the start and end events.

By default, a Detail report displays all statistics for all items, associated with every occurrence of every event during the collection period, from every facility in the formatted Oracle Rdb database. Detail reports can be very large. Therefore, Oracle Corporation recommends that you use the /FACILITY, /EVENT, and /ITEM qualifiers of the REPORT command to limit Detail reports to specific facilities, events, and items. You can further limit Detail reports by using the /OPTIONS qualifier as described in Chapter 4. You can also use the /SINCE, /BEFORE, or /DURATION qualifiers to report on a portion of a collection period.

You use a command similar to the following if you want to produce a Detail report on the ATM_SAMPLE balance event:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=DETAIL -
 _$/FAC=(ATM_SAMPLE) -
 _$/EVENT=BALANCE_EVENT -
 _$/OUTPUT=SAMPLE_DETAIL_REPORT.RPT
```

Example 3–8 shows part of the Detail report the previous command produces.

Example 3–8 Detail Report Segment Showing Balance Event Items

```
25-AUG-1994 20:31                Detail Report                          Page 1
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                        14

22-JUL-1994 17:45:47.49       4.53        42759        27403            2

22-JUL-1994 17:45:52.02                   42789        27481            2

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                    IOs         SIZE

     8        17530        67552         3062         7988          251

     9        17601        68368         3097        11219          674

 PRIVATE WS      WORKING
                 SET SIZ

       1248         2469

       2203         4005

            ================= Next Occurrence ====================
```

(continued on next page)

Example 3–8 (Cont.) Detail Report Segment Showing Balance Event Items

```
Timestamp                 Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                      14
22-JUL-1994 17:46:36.47      1.93        44139        27755           2
22-JUL-1994 17:46:38.40                  44152        27757           2

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        17745        68601         3097        11547         783
     9        17745        68601         3097        11547         783

 PRIVATE WS      WORKING
                 SET SIZ

       2262         4517
       2262         4517

             ================= Next Occurrence ====================

Timestamp                 Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                      14
22-JUL-1994 17:47:37.83      1.88        46108        28123           2
22-JUL-1994 17:47:39.71                  46121        28126           2

25-AUG-1994 20:31               Detail Report                      Page 2
Selection: SAMPLE_SELECTION                              Oracle Trace V2.2

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        17947        68664         3097        11547         783
     9        17947        68664         3097        11547         783

 PRIVATE WS      WORKING
                 SET SIZ

       2325         4517
       2325         4517

             ================= Next Occurrence ====================

     .
     .
     .
```

The facility definition associated with the collection defines the characteristics of item headers, such as CURRENT_PRIO, DIRECT_IO, and PAGE_FAULTS. You can override the default header text and width using the ITEMS keyword to the /OPTIONS qualifier described in Chapter 4.

3.6 Frequency Report

The Frequency report is useful for tracing the execution of an instrumented application. The Frequency report displays the number of event occurrences for each event that matches the selection criteria specified on the REPORT command. The layout of the report is similar to the Summary report (see Section 3.3) except that it displays the total occurrences of an event during a time period, rather than the statistics associated with the event.

The /INTERVAL qualifier of the REPORT command pertains only to Frequency reports. This qualifier specifies the interval for which Oracle Trace counts event occurrences. The keywords to the /INTERVAL qualifier, SECOND, MINUTE, and HOUR, correspond to the intervals they specify. For example, the SECOND keyword specifies an interval of one second. The default is an interval of one minute.

For duration events, Oracle Trace counts only occurrences of completed start and end pairs and displays a diagnostic message for incomplete pairs.

Example 3–9 shows a Frequency report based on data collected from the ATM sample application. The following example shows the command to generate the report:

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/TYPE=FREQUENCY -
 _$/OUTPUT=SAMPLE_FREQUENCY_REPORT.RPT
```

Although the Frequency report provides the least information of the three report types, it is useful for tracing the execution of an instrumented application. For example, the ATM report (Example 3–9) shows that the balance event occurs much more frequently than the other events. By examining the timestamps on the events (perhaps by generating a new report using /INTERVAL=SECONDS), it becomes evident that customers are checking their account balances before and after each transaction. Based on this information, the application programmer might choose to modify the ATM program to automatically display the new balance after each transaction.

Example 3–9 Sample Frequency Report Based on ATM Data

```
28-SEP-1994 16:06              Frequency Report                        Page 1
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Time Period          Occurrences
24-SEP-1994 17:00:00          19
24-SEP-1994 17:01:00          13
24-SEP-1994 17:02:00          19
24-SEP-1994 17:03:00          17
24-SEP-1994 17:04:00          12
24-SEP-1994 17:05:00          13
28-SEP-1994 16:06              Frequency Report                        Page 2
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Deposits        In Facility:  ATM_SAMPLE      Version:  V2.2

Time Period          Occurrences

24-SEP-1994 17:01:00          12
24-SEP-1994 17:02:00           7
24-SEP-1994 17:03:00           4
24-SEP-1994 17:05:00           8

%EPC-I-NOEND, 1 Start Event Records had no matching End

28-SEP-1994 16:06              Frequency Report                        Page 3
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Time Period          Occurrences

24-SEP-1994 17:00:00           5
24-SEP-1994 17:03:00          10
24-SEP-1994 17:04:00          13
28-SEP-1994 16:06                    Index                             Page 9
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Report Index

Facility Name               Event Name                       Join item  Page

ATM_SAMPLE                  Balance                                        1
ATM_SAMPLE                  Deposits                                       2
ATM_SAMPLE                  Withdrawals                                    5
```

Note in Example 3–9 that because no withdraw events occurred during the 24-SEP-1994 17:01:00 and 24-SEP-1994 17:02:00 intervals, Oracle Trace did not report these intervals.

3.7 How Oracle Trace Wraps Text in Reports

Oracle Trace wraps text in reports when a name or a value overflows the space allotted for it. Table 3–3 describes how Oracle Trace handles wrapping for each kind of report part.

Table 3–3 Text Wrapping in Oracle Trace Reports

Report Part	Where It Wraps	How It Wraps
Groupable Item Header[1]	At the last report column.	Continues on the next line without indenting. See Example 3–10 for an example.
Header Text	At the last underscore or space. If header does not contain underscores or spaces, the header wraps at the last column of the header field.	Continues to the next line and right-justifies within the header column. See Example 3–12 for an example.
Items	At the column where the last whole report item fits on the line.	Continues as a set of rows. See Example 3–11 for an example.
Item Text	At the last column of the field in which it occurs.	Continues to the next line and left-justifies within the column. See Example 3–13 for an example.

[1]The Groupable Item Heads appear only in reports produced using the /GROUP_BY qualifier.

Example 3–10 shows a portion of a report in which a long file name wraps.

Example 3–10 Wrapping of a Groupable Item

```
27-MAY-1994 21:45            The Longest Files                        Page 1
Selection: ACC_TRENDS                                      Oracle Trace V2.2

Event TRANSACTION  In Facility RDBVMS                  Version V3.1-0
  For Collections 23MAYAM,24MAYAM
  For Nodes MYVAX1, MYVAX2
  With STREAM_ID = 392
  And  STREAM_ID = 393

DB Name: THIS_IS_A_VERY_LONG_DEVICE_NAME_LOGICAL:[DIRECTORY_LEVEL
         _ONE.DIRECTORY_LEVEL_TWO.DIRECTORY_LEVEL_THREE]THE_DATABASE_FILE
         _NAME.RDB
```

(continued on next page)

Example 3–10 (Cont.) Wrapping of a Groupable Item

```
   .
   .
   .
```

Example 3–11 shows how groups wrap in a report.

Example 3–11 Wrapping Items

```
27-MAY-1994 21:45        Monday AM Accounting Transactions             Page 1
Selection: ACC_TRENDS                                        Oracle Trace V2.2

Event TRANSACTION  In Facility RDBVMS                 Version V3.1-0
  For Collections 23MAYAM,24MAYAM
  For Nodes MYVAX1, MYVAX2
  With STREAM_ID = 392
  And  STREAM_ID = 393

Client PC: 3578

           Elapsed   CPU TIME  BUFFERED IO  DIRECT IO  PRIVATE WS  GLOBAL WS

Minimum       0.02          2           25        685         325        287
Maximum      10.65        534          945       1052        2056       4912
Count          346

          RUJ File     AIJ File      Locks
             Reads       Writes  Requested
Minimum         65          12         39
Maximum         75          17       1985
Count          346

Client PC: 3580

           Elapsed   CPU TIME  BUFFERED IO  DIRECT IO  PRIVATE WS  GLOBAL WS

   .
   .
   .
```

Example 3–12 shows a portion of a database transaction report where the header text wraps.

Example 3–12 Wrapping of Header Text

```
27-MAY-1994 21:45        Monday AM Accounting Transactions              Page 1
Selection: ACC_TRENDS                                          Oracle Trace V2.2

Event TRANSACTION  In Facility RDBVMS                  Version V3.1-0
  For Collections 23MAYAM,24MAYAM
  For Nodes MYVAX1, MYVAX2
  With STREAM_ID = 392
  And  STREAM_ID = 393

Client PC: 3578

         Pagefault  Pagefault  Free_VM    Lock Reqs  WS Size    WS Global
                s        _IO    Bytes
Minimum  . . .
Maximum  . . .
Count    . . .
   .
   .
   .
```

Example 3–13 shows a portion of a report where the item text wraps.

Example 3–13 Wrapping of Item Text

```
27-MAY-1994 21:45          Accounting Database Binds                  Page 1
Selection: ACC_TRENDS                                          Oracle Trace V2.2

Event DATABASE  In Facility RDBVMS                     Version V3.1-0
  In Collections 23MAYAM,24MAYAM
  For Nodes MYVAX1, MYVAX2

Point Timestamp          DB Name       Stream Id  Client PC

20-Apr-1994 09:22:45.67  ACC_DISK:[DA       3785     126457
                         TABASES]ACCO
                         UNTING_MAIN.
                         RDB
20-Apr-1994 18:34:23.62  ACC_DISK:[DA       3782     129381
                         TABASES]ACCO
                         UNTING_NEW.R
                         DB
   .
   .
   .
```

4

Producing Customized Oracle Trace Reports

As described in Chapter 3, the Oracle Trace Summary, Detail, and Frequency reports provide information about events. The /EVENTS qualifier gives you some ability to control event information in these reports. For example, you can use the /EVENTS qualifier to limit reports to specific events, and to specify the order in which the reports display events. If you want more control over event information in reports, you can use the /OPTIONS qualifier, which is mutually exclusive with the /EVENTS qualifier. For example, with the /OPTIONS qualifier, you can:

- Create reports with unique information for each event
- Specify the items you want reported for each event
- Limit the data reported for each event, based on specific characteristics, such as:
  - Collection name
  - Process identifier (EPID)
  - Image name
  - Item name
  - Item value
  - Node name
- Override display characteristics defined in the facility definition for items

For example, you can create a report with a unique subreport for each event. Each subreport pertains to one event and can have its own title, be a unique report type, and limit data based on its own criteria.

Note

Although the /OPTIONS qualifier allows you to specify many events with unique characteristics in a single report, you cannot specify the same event more than once per report. Therefore, if you want

to produce more than one report for an event, you must specify two separate reports. For example, to produce Summary and Detail reports for the DEPOSIT_EVENT, you enter the following report commands:

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB/OPTIONS
_$ EVENT DEPOSIT_EVENT -
_$ /FACILITY=ATM_SAMPLE-
_$ /TYPE=FREQUENCY


$ COLLECT REPORT FORMATTED_SAMPLE.RDB/OPTIONS
_$ EVENT DEPOSIT_EVENT -
_$ /FACILITY=ATM_SAMPLE-
_$ /TYPE=DETAIL
```

4.1 Keywords to the /OPTIONS Qualifier

The /OPTIONS qualifier has keywords for specifying report characteristics. These keywords are:

- EVENT—Denotes the beginning of an event-specific subreport and names the event to which it pertains. All subsequent keywords pertain to this event, until you specify the EVENT keyword again.
- ITEM—Allows you to override the display characteristics that are defined in the facility definition for items associated with this event.
- JOIN—Allows you to join events from two or more event-specific subreports based on a shared item value.
- RESTRICTION — Allows you to limit the event report to data that meets the criteria that you specify.

4.2 Using the /OPTIONS Qualifier

You can use the /OPTIONS qualifier to the REPORT command in two ways:

- By typing /OPTIONS on the REPORT command line and entering keywords and keyword qualifiers interactively in response to the OPTIONS> prompt, as shown in the following example:

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB -
$_/OUTPUT=CUSTOMIZED_REPORT.RPT -
$_/OPTIONS
Options> EVENT DEPOSIT_EVENT -
Options> /FACILITY=ATM_SAMPLE -
Options> /TYPE=FREQUENCY -
Options> /SUBTITLE="Deposit Event Frequency"
Options> EVENT WITHDRAWAL_EVENT -
Options> /FACILITY=ATM_SAMPLE -
Options> /TYPE=SUMMARY -
Options> /SUBTITLE="Withdrawal Event Summary"
   .
   .
   .
Options> CTRL/Z
```

- By creating an options file containing keywords and keyword qualifiers (see Example 4–1), and supplying the file name as a value to the /OPTIONS qualifier, as follows:

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB -
 $_/OUTPUT=CUSTOMIZED_REPORT.RPT -
 $_/OPTIONS=OPTIONS.OPT
```

Example 4–1 OPTIONS.OPT Example Options File

```
EVENT DEPOSIT_EVENT-
        /FACILITY=ATM_SAMPLE-
        /TYPE=FREQUENCY
!
EVENT WITHDRAWAL_EVENT-
        /FACILITY=ATM_SAMPLE-
        /TYPE=SUMMARY-
        /SUBTITLE="Deposit Event Summary"
EVENT TRANSACTION-
        /FACILITY=RDBVMS-
        /TYPE=SUMMARY
JOIN ATM_SAMPLE.DEPOSIT_EVENT.CROSS_FAC_14=RDBVMS.TRANSACTION.CROSS_FAC_14
 RESTRICTION NODE OOTOOL
!
EVENT SIGNIN_EVENT-
        /FACILITY=ATM_SAMPLE-
        /TYPE=DETAIL-
        /SUBTITLE="One Minute Deposit Event Detail"-
        /SINCE="22-jul-1994 17:47"-
        /BEFORE="22-jul-1994 17:48"
!
```

In both cases, you must specify each keyword on a separate line. See Chapter 7 for more information about the /OPTIONS qualifier and its keywords.

The /OPTIONS qualifier is also useful when your Oracle Trace command is greater than the 255-character limit. Using the /OPTIONS qualifier lets you shorten the command by putting some of the command keywords and qualifiers in the options file.

4.3 Example of a Customized Report

The examples in this section contain:

- Segments of an example command that produce a customized report
- Portions of the report that the command produces.

Example 4–2 is an example of the REPORT command that specifies default characteristics for the whole report and characteristics for the first subreport.

Example 4–2 Command Segment Specifying the First Subreport

```
$COLLECT REPORT FORMATTED_SAMPLE.RDB -
 _$/OUTPUT=SPECIAL_REPORT.RPT -
 _$/OPTIONS
 _$EVENT DEPOSIT_EVENT ❶ -
 _$/FACILITY=ATM_SAMPLE -
 _$/TYPE=FREQUENCY ❷ -
 _$/SUBTITLE="Deposit Event Frequency Subreport" -
    .
    .
    .
```

❶ Specifies the first event-specific subreport for the Deposit Event.

❷ Specifies a Frequency report.

Example 4–3, is an example of the first subreport, produced by the command in Example 4–2.

❶ Frequency report header.

❷ Event-specific subreport subtitle.

❸ Subreport pertains to Deposits.

Example 4–3 First Event-Specific Subreport Produced by the Command in Example 4–2

```
31-AUG-1994 16:07              Frequency Report ❶                Page 1
Selection: SAMPLE_SELECTION                                Oracle Trace V2.2

                        "Deposit Event Frequency Subreport"❷

Event:  Deposits ❸      In Facility:  ATM_SAMPLE   Version:  V2.2

Time Period          Occurrences

22-JUL-1994 17:46:00          28
22-JUL-1994 17:47:00           1
22-JUL-1994 17:48:00          27
22-JUL-1994 17:49:00          30
❹
22-JUL-1994 17:51:00           1
22-JUL-1994 17:53:00          11
22-JUL-1994 17:54:00           7
22-JUL-1994 17:55:00          18
   .
   .
   .
```

❹ No Deposit events occurred during the 22-JUL-1994 17:50:00 interval. Therefore, Oracle Trace did not report this interval. Note that because Example 4–2 did not specify the /INTERVAL qualifier, the report uses the default of minute.

Example 4–4 is an example of the command segment that specifies the second subreport.

Example 4–4 Command Segment Specifying the Second Subreport

```
   .
   .
   .
$EVENT WITHDRAW_EVENT  ❶ -
 _$/FACILITY=ATM_SAMPLE ❷ -
 _$/TYPE=SUMMARY ❸ -
 _$/STATISTICS=ALL ❹ -
 _$/SUBTITLE="Withdrawal Event Summary" ❺ -
   .
   .
   .
```

❶ Specifies this event-specific subreport for the Withdrawal Event.

❷ Limits the report to the ATM_SAMPLE facility.

❸ Specifies a Summary report.

❹ Specifies all statistics.

❺ Specifies the subtitle "Withdrawal Event Summary".

Example 4–5 is an example of the second subreport specified by the command in Example 4–4.

Example 4–5 Second Subreport Example Produced by the Command in Example 4–4

```
   .
   .
   .
1-NOV-1994 12:04                Summary Report ❶                  Page 1
Selection: SAMPLE_SELECTION                                Oracle Trace V2.2

                                Withdrawal Event Summary ❷

Event:  Withdrawals ❸   In Facility:  ATM_SAMPLE ❹     Version:  V2.2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN    DIRECT IO
                                                       14  T PRIO

Minimum       0.38          13            2           4       8            4
Maximum       6.16          15            9           4       9           10
Mean          0.98       13.00         5.24        4.00    8.98         4.64
Std Dev       0.88        0.12         1.37        0.00    0.12         0.75
95 Prct       2.72       13.25         7.93        4.00    9.23         6.13
Total       246.84        3252         1310        1000    2246         1162
Count          250

          PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                              IOs         SIZE

Minimum            0           0        10853         668         2174
Maximum           63           0        10857         675         2245
Mean            0.27        0.00     10856.84      674.50      2241.95
Std Dev         3.98        0.00         0.78        1.44         6.51
95 Prct         8.08        0.00     10858.37      677.34      2254.71
Total             68           0      2714210      168627       560489
Count            250

            WORKING
            SET SIZ

Minimum        4005
Maximum        4005
Mean        4005.00
Std Dev        0.00
95 Prct     4005.00
Total       1001250
Count           250

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

❶ Summary Report Header.

❷ Subreport subtitle.

❸ Subreport pertains to the Withdrawal Event.

❹ Report is limited to Withdrawal events in the ATM_SAMPLE facility.

Example 4–6 is a command segment that specifies the third event-specific subreport.

Example 4–6 Command Specifying the Third Subreport

```
   .
   .
   .
$EVENT BALANCE_EVENT -
 _$/FACILITY=ATM_SAMPLE -
 _$/TYPE=DETAIL ❶ -
 _$/SUBTITLE="Two-Minute Balance Event Detail" ❷ -
 _$/SINCE="22-jul-1994 17:47" ❸ -
 _$/BEFORE="22-jul-1994 17:49" -
   .
   .
   .
```

❶ Specifies a Detail report.

❷ Subreport subtitle is "Two-Minute Balance Event Detail".

❸ Limits the report period to between 17:47 and 17:49 p.m. of July 22, 1994.

Example 4–7 is an example of the third subreport produced by the command segment in Example 4–6.

Example 4–7 Third Subreport Example Produced by the Command in Example 4–6

```
   .
   .
   .
31-AUG-1994 16:00                Detail Report ❶                    Page 1
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

                        Two-Minute Balance Event Detail ❷

Event:  Balances ❸       In Facility:  ATM_SAMPLE    Version:  V2.2

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                        14

22-JUL-1994 17:47:41.98       3.80        46151        28130           3

22-JUL-1994 17:47:45.78                   46165        28134           3

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        17947        68664         3097        11547          783

     5        17953        68664         3097        11547          783

 PRIVATE WS      WORKING
                 SET SIZ

       2325         4517

       2325         4517

             ================= Next Occurrence ====================

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                        14

22-JUL-1994 17:48:03.72       1.30        46619        28212           3

22-JUL-1994 17:48:05.02                   46633        28217           3

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        17974        68667         3097        11547          786

     5        17979        68668         3097        11547          787
   .
   .
   .
```

(continued on next page)

Example 4–7 (Cont.) Third Subreport Example Produced by the Command in Example 4–6

```
            ================= Next Occurrence ====================

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                       14

22-JUL-1994 17:48:09.44       1.39        46794        28241            3

22-JUL-1994 17:48:10.83                   46808        28246            3

31-AUG-1994 16:00                Detail Report                      Page 2
Selection: SAMPLE_SELECTION                                 Oracle Trace V2.2

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        17984        68668         3097        11547          787

     4        17990        68669         3097        11547          787
   .
   .
   .
            ================= Next Occurrence ====================

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                       14

22-JUL-1994 17:48:11.35       0.58        46838        28249            3

22-JUL-1994 17:48:11.93                   46852        28254            3
   .
   .
   .
            ================= Next Occurrence ====================

Timestamp                  Elapsed  BUFFERED IO     CPU TIME    CROSS FAC
                                                                       14

22-JUL-1994 17:48:42.61       0.52        47894        28451            3

22-JUL-1994 17:48:43.13                   47908        28456            3

CURREN    DIRECT IO   PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS
T PRIO                                   IOs         SIZE

     8        18101        68670         3097        11547          788

     4        18105        68670         3097        11547          788
```

(continued on next page)

Example 4–7 (Cont.) Third Subreport Example Produced by the Command in Example 4–6

```
   .
   .
   .
```

❶ Detail report.

❷ Subreport title "Two-Minute Balance Event Detail Report".

❸ Balance event.

Example 4–8 is a command segment that specifies the fourth event-specific subreport.

Example 4–8 Fourth Subreport Command

```
   .
   .
   .
$EVENT SIGNIN_EVENT ❶ ❷ -
 _$/FACILITY=ATM_SAMPLE -
 _$/SUBTITLE="Signin Change Item Width" ❸ -
 _$ITEM MACHINE_NUMBER ❹ -
 _$/WIDTH=7 ❺ -
   .
   .
   .
```

❶ Specifies the Signin event for this subreport.

❷ Specifies a Summary report by default.

❸ Specifies a subtitle of "Signin Change Item Width".

❹ Changes display characteristic specified in the facility definition for the MACHINE_NUMBER item.

❺ Specifies a column width of 7 for the MACHINE_NUMBER item.

Example 4–9 is an example of the fourth subreport.

Example 4–9 Fourth Subreport Example Produced by the Command in Example 4–8

```
   .
   .
   .
31-AUG-1994 16:03              Summary Report ❶                  Page 1
Selection: SAMPLE_SELECTION                                Oracle Trace V2.2

                          Signin Change Item Width ❷

Event:  Signins ❸       In Facility:  ATM_SAMPLE   Version:  V2.2

          Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                      14  T PRIO

Count           1

         Machine ❹   PAGEFAULTS    PAGEFAULT      VIRTUAL   GLOBAL WS
          Number                      IOs         SIZE

Count          1

          PRIVATE WS      WORKING
                          SET SIZ

Count             1

%EPC-I-NOSTART, 2 End Event Records had no matching Start  ❺
   .
   .
   .
```

❶ Summary report specified by default because neither the REPORT command nor the EVENT keyword specified the /TYPE qualifier.

❷ Subreport title.

❸ Subreport pertains to the Signin event.

❹ A column-width of 7 overrides the facility definition value for the column width of the MACHINE_NUMBER item.

❺ During this collection period, two event records had no matching EPC$START_EVENT(W) and EPC$END_EVENT(W) routine call pairs.

4.4 Connecting Events Using the JOIN Keyword

You can relate events and items within a single facility and across facilities using the RELATE keyword of the CREATE DEFINITION command. You can also use the /JOIN qualifier of the REPORT command to produce reports that connect events and items based on the relationships you define with the RELATE keyword. Likewise, you can use the JOIN keyword of the /OPTIONS qualifier to connect items across subreports based on a shared value of that item.

Note

Oracle Corporation recommends that you read Chapter 5 if you plan to use the JOIN keyword.

The command in Example 4–10 creates a subreport in which the JOIN keyword connects the ATM_SAMPLE's WITHDRAW_EVENT and BALANCE_EVENT based in the ACCOUNT_ID value they share.

Example 4–10 Example Command Connecting Two Events

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB -
 $/OUTPUT=JOINED_SUBREPORTS.RPT -
 _$/TYPE=SUMMARY -
 _$/STATISTICS=ALL -
 _$/OPTIONS
 OPTIONS> EVENT WITHDRAW_EVENT -
 OPTIONS> /FACILITY=ATM_SAMPLE
 OPTIONS> EVENT BALANCE_EVENT -
 OPTIONS> /FACILITY=ATM_SAMPLE
 OPTIONS> JOIN ❶ ATM_SAMPLE.WITHDRAW_EVENT.ACCOUNT_ID= -
 OPTIONS> ATM_SAMPLE.BALANCE_EVENT.ACCOUNT_ID
```

❶ The JOIN connects events across subreports.

Example 4–11 is an example report that this command produces.

Example 4–11 Example Report with Two Connected Events

```
24-SEP-1994 18:32                Summary Report                          Page 1
Selection: SAMPLE_SELECTION                                    Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 1

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO

Minimum       0.42          13            4           4       9           4
Maximum       3.52          13            9           4       9           5
Mean          2.28       13.00         6.25        4.00    9.00        4.58
Std Dev       0.82        0.00         1.35        0.00    0.00        0.51
95 Prct       2.89       13.00         8.90        4.00    9.00        5.59
Total        15.41         156           75          48     108          55
Count           12

          PAGEFAULTS    PAGEFAULT      VIRTUAL    GLOBAL WS   PRIVATE WS
                              IOs         SIZE

Minimum            0           0        10853         668         2174
Maximum            1           0        10857         675         2245
Mean            0.16        0.00     10853.66      668.91      2230.66
Std Dev         0.38        0.00         1.55        2.23        18.08
95 Prct         0.92        0.00     10856.71      673.29      2266.10
Total              2           0       130244        8027        26768
Count             12

            WORKING
            SET SIZ

Minimum        4005
Maximum        4005
Mean        4005.00
Std Dev        0.00
95 Prct     4005.00
Total         48060
Count            12

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

24-SEP-1994 18:32                Summary Report                          Page 2
Selection: SAMPLE_SELECTION                                    Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 1

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                       14  T PRIO
```

(continued on next page)

Example 4–11 (Cont.) Example Report with Two Connected Events

```
Minimum       0.45          13           3           2       9           0
Maximum       8.28          13          13           2       9           7
Mean          2.17       13.00        4.78        2.00    9.00        0.57
Std Dev       2.20        0.00        2.51        0.00    0.00        1.86
95 Prct       6.49       13.00        9.71        2.00    9.00        4.23
Total        30.46         182          67          28     126           8
Count           14

   .
   .
   .
24-SEP-1994 18:32                Summary Report                        Page 3
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Withdrawals     In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                      14  T PRIO

Minimum       0.38          13           2           4       8           4
Maximum       6.16          15           9           4       9          10
Mean          0.97       13.00        5.18        4.00    8.98        4.65
Std Dev       0.88        0.12        1.35        0.00    0.12        0.76
95 Prct       2.71       13.26        7.84        4.00    9.23        6.15
Total       231.43        3096        1235         952    2138        1107
Count          238

   .
   .
   .
24-SEP-1994 18:32                Summary Report                        Page 4
Selection: SAMPLE_SELECTION                                  Oracle Trace V2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

Join Level =   1
Account Number : 2

           Elapsed  BUFFERED IO     CPU TIME    CROSS FAC  CURREN   DIRECT IO
                                                      14  T PRIO
```

(continued on next page)

Example 4–11 (Cont.) Example Report with Two Connected Events

```
Minimum       0.24          13            1            2       8           0
Maximum       7.91          13            6            2       9           1
Mean          0.97       13.00         3.47         2.00    8.99        0.01
Std Dev       1.15        0.00         0.92         0.00    0.08        0.10
95 Prct       3.23       13.00         5.28         2.00    9.15        0.20
Total       286.55        3835         1025          590    2653           3
Count          295
   .
   .
   .

24-SEP-1994 18:32                    Index                            Page 5
Selection: SAMPLE_SELECTION                                 Oracle Trace V2.2

Report Index

Facility Name               Event Name                    Join item  Page

ATM_SAMPLE                  Withdrawals                     1          1
ATM_SAMPLE                  Balance                         1          2
ATM_SAMPLE                  Withdrawals                     2          3
ATM_SAMPLE                  Balance                         2          4
```

5
Optimizing the Report Process

This chapter describes ways you can optimize the report process.

5.1 How to Optimize Reporting from Large Formatted Databases

If you are generating reports from large, formatted databases, you can improve reporting performance by applying these recommendations:

- You can put your database journal file on another device by assigning the logical RDMS$RUJ. For example:

  ```
  $ DEFINE RDMS$RUJ DISK5:[TEMPFILES]
  ```

- You can put your SORT work files on another device by assigning the logicals SORTWORK0 and SORTWORK1. For example:

  ```
  $ ASSIGN DISK1: SORTWORK0
  $ ASSIGN DISK2: SORTWORK1
  ```

- You may need to increase the enqueue limit (ENQLM) quota for your process. The ENQLM quota allows you to limit the number of locks that a process may take out. Oracle Corporation recommends a minimum ENQLM value of 1800. If this value is too low, you will get the following error:

  ```
  %EPC-E-Bugcheck, Fatal error
  %NONAME-F-NOMGG,.............
  -RDMS-F-EXQUOTA, exceeded Quota
  -SYSTEM-F-EXENQLM, exceeded Enque Quota
  ```

  To generate a report from a 300K block formatted database, you may need to raise your ENQLM to 10000. Note that increasing the ENQLM has no negative effects on the system in terms of preallocated resources. Also note that you will have to log out and back in again for the change to take effect. See the OpenVMS documentation for information on changing process quotas.

5.2 Optimizing Reports That Use the /JOIN Qualifier

You can improve reporting performance for reports that use the /JOIN qualifier by using an Oracle Rdb index. This procedure is most effective if you perform the same report repeatedly.

To perform this procedure, you must have write access to the formatted database. If you do not, use the SET ACCESS command to acquire it.

Perform the actions associated with the steps in Table 5–1 to identify and produce a report using an index.

Table 5–1 How to Identify and Use an Index

Step	Action

1	Determine the index to define by:
	a. Defining the strategy and BLR (binary language representation) Oracle Rdb debug flags, for example:
	`$ DEFINE RDMS$DEBUG_FLAGS SB`
	b. Running your Oracle Trace REPORT command, for example:
	`$ COLLECT REPORT ATM -`
	`_$/TYPE=SUMMARY/STATISTICS=ALL/FACILITY=(ATM_SAMPLE,RDBVMS) -`
	`_$/EVENTS=(withdraw_event,transaction) -`
	`_$/OUT=REPORT.RPT -`
	`_$/JOIN=(ATM_SAMPLE.WITHDRAW_EVENT.CROSS_FAC_14= -`
	`_$RDBVMS.TRANSACTION.CROSS_FAC_14)`
	The screen produces a display as shown in Example 5–1

2	Stop the report when the table and field names display to save time, for example: Ctrl/y

(continued on next page)

Table 5–1 (Cont.) How to Identify and Use an Index

Step	Action

3	Define the index, for example:
	`$ SQL :== $SQL$`
	`$ SQL`
	`DECLARE SCHEMA FILE ATM;`
	`CREATE INDEX WITHDRAW_INDEX1 ON EPC$1_4094_WITHDRAW_EVENT`
	`(CROSS_FAC_14_END)`
	`TYPE IS SORTED;`
	`COMMIT;`
	`FINISH;`
	`EXIT`
	`$`

4	Define the Oracle Rdb strategy debug flag, for example: `$ DEFINE RDMS$DEBUG_FLAGS S`

5	Reenter the Oracle Trace command, for example:
	`$ COLLECT REPORT ATM -`
	`_$/TYPE=SUMMARY/STATISTICS=ALL/FACILITY=(ATM_SAMPLE,RDBVMS) -`
	`_$/EVENTS=(withdraw_event,transaction) -`
	`_$/OUT=REPORT.RPT -`
	`_$/JOIN=(ATM_SAMPLE.WITHDRAW_EVENT.CROSS_FAC_14= -`
	`_$RDBVMS.TRANSACTION.CROSS_FAC_14)`

5.3 Defining an Index

If you have interactive SQL or interactive RDO on your system and there are many instances of data in the database, you can speed up report generation by adding an index to a relation. Adding an index to the REQUEST relation of the Oracle CODASYL DBMS in the formatted database is especially advantageous if you perform a /GROUP_BY and RESTRICT for Oracle CODASYL DBMS.

The SQL syntax to do this is:

```
SQL> CREATE INDEX MY_INDEX ON EPC$1_40_REQUEST
cont> (IMAGE_RECORD_ID, CLIENT_PC);
SQL> COMMIT;
```

Example 5–1 Oracle Rdb Debug Display

```
0000 (00000) BLR$K_VERSION4
0001 (00001) | BLR$K_BEGIN
0002 (00002) | BLR$K_MESSAGE 1 137
0006 (00006) | | DSC$K_DTYPE_L 0
0008 (00008) | | DSC$K_DTYPE_W 0
000A (00010) | | DSC$K_DTYPE_L 0
   .
   .
   .
07A0 (01952) | | | BLR$K_GROUP_VALUE  67
07A2 (01954) | | BLR$K_BOOLEAN
07A3 (01955) | | | BLR$K_ANY2
07A4 (01956) | | | BLR$K_EQL
07A5 (01957) | | | BLR$K_FIELD_ID 1 0
07A9 (01961) | | | BLR$K_RSE  1
07AB (01963) | | | |  BLR$K_RELATION EPC$1_4094_WITHDRAW_EVENT 3 ❶
07C7 (01991) | | | | BLR$K_PROJECT  1
07C9 (01993) | | | | BLR$K_FIELD 3 CROSS_FAC_14_END ❷
07DC (02012) | | | BLR$K_END
07DD (02013) | | | BLR$K_FIELD 3 CROSS_FAC_14_END  ❸
07F0 (02032) | | BLR$K_SORT  2
```

❶ Table name

❷ Field name index

❸ Field name index

If you have interactive RDO, the syntax is:

```
RDO> DEFINE INDEX MY_INDEX FOR EPC$1_40_REQUEST_ACTUAL
RDO> TYPE IS SORTED.
RDO> STREAM_ID ASCENDING.
RDO> CLIENT_PC ASCENDING.
RDO> END MY_INDEX INDEX.
RDO> COMMIT;
```

Note that adding an index will cause a delay in merging another data capture file into this formatted database, and the index will not help if you want to use GROUP_BY with a different set of items. You can drop the index before merging the databases by entering either of the following sets of commands:

```
SQL> DROP INDEX MY_INDEX;
SQL> COMMIT;
```

or,

```
RDO> DELETE INDEX MY_INDEX;
RDO> COMMIT;
```

See Appendix A for a description of the relations in the formatted database and product specific documentation for more information.

6

Troubleshooting Oracle Trace Reports

This chapter describes some of the problems you might encounter when you produce Oracle Trace reports. Table 6–1 shows the symptoms, causes, and corrections of common reporting errors.

Table 6–1 Common Reporting Errors

Report Type	Symptom	Cause	Correction
All	%DCL-W-TKNOVF, command element is too long - shorten	Report fails because there are too many characters in the report command. A command cannot exceed 255 characters.	Create an options file and use the /OPTIONS qualifier as described in Section 4.2.
All	%EPC-E-xxxxx.	Report command with /OPTIONS qualifier fails.	Create an options file to specify keyword options and qualifiers, as described in Section 4.2. Enter the OpenVMS SET VERIFY command before you enter the REPORT command to help you determine where the command is failing.
All	%EPC-I-RPQU_ NODATFND, No Data Found For Report in report file.	Events for JOIN incorrectly specified.	See Section 6.2.

(continued on next page)

Table 6–1 (Cont.) Common Reporting Errors

Report Type	Symptom	Cause	Correction
All	ACMS reports display ACMS$PROCEDURE <procedure_index> in place of ACMS procedure names.	The ACMS task group (.TBD) file was built prior to ACMS Version 3.3	Rebuild the task group using a command similar to the following: `$ ADU BUILD GROUP task_group_name`.
Detail	Hyphens (- - - -) fill the elapsed-time field for a duration event; however, other item values appear correct.	The system clock was changed while the collection was running.	No correction is necessary because the values are not affected. You can rerun the report if you wish.
	Hyphens (- - - -) fill the elapsed-time field for a duration event, and values for other items are either zeros or negative.	The EPC$START_EVENT system service call in your application is incorrectly placed after the event, and the EPC$END_EVENT call is placed at the start of the event.	Modify your application to correct the placement of the EPC$START_EVENT and EPC$END_EVENT calls.
Summary	An event is not reported and there is a message at the end of the report displaying the number of occurrences of end-time stamps preceding start-time stamps.	The system clock was changed while the collection was running.	No correction is necessary because item values are not affected. You can rerun the report if you wish.

(continued on next page)

Table 6–1 (Cont.) Common Reporting Errors

Report Type	Symptom	Cause	Correction
		The EPC$START_EVENT system service call in your application is incorrectly placed after the event, and the EPC$START_EVENT call is placed at the start of the event.	Modify your application to correct the placement of the EPC$START_EVENT and EPC$END_EVENT calls.
Detail and Summary	Hyphens (- - - -) appear in an item value field.	The item is incorrectly defined in the facility definition as a type counter, and does not exist in the item list for both the start and end events.	Modify the facility definition so that the item appears on the item list for both start and end events.
	Asterisks (****) fill an item value field.	The field width is too small to accommodate the value.	See Section 6.1.
Summary, Detail, and Frequency	Percent signs (%%%%) fill a value field.	An arithmetic overflow error occurred during the report.	Subdivide the report using the /GROUP_BY qualifier of the REPORT command. This will provide correct line-item values. However, the overflow error might still occur on the grand total.

6.1 Field Width Too Small for Value

When the field width is too small for a value, Oracle Trace reports display a row of asterisks in the field. When this happens, specify a larger width for the field using the /OPTION qualifier as follows:

```
$ COLLECT REPORT MY_DATABASE /OPTIONS
Options>EVENT MY_EVENT /FACILITY=MY_FACILITY
Options>ITEM BIG_VALUE /WIDTH=32
Options>CTRL/Z
$
```

If this happens in a report that you run often on an application you have instrumented, you might consider changing the field width for the item in the facility definition.

Table 6–2 lists the Summary report statistics and their format.

Table 6–2 The Format of Summary Report Statistics

Statistic	Elapsed Time	Others
Minimum	ZZZZZ9.99	ZZZZZZZZ9
Maximum	ZZZZZ9.99	ZZZZZZZZ9
Mean	ZZZZZ9.99	ZZZZZ9.99
Std Dev[1]	ZZZZZ9.99	ZZZZZ9.99
95 Prct[2]	ZZZZZ9.99	ZZZZZ9.99
Total	ZZZZZZZZ9	ZZZZZZZZ9
Count[3]	ZZZZZZZZ9	

[1]Std Dev is Standard Deviation.

[2]95 Prct is 95th Percentile.

[3]Because the Count is the same for all report items, it is only displayed for the first report item.

Z—represents a numeric character 0 through 9 for which leading zeros are replaced with blanks. Negative values display a minus sign to the left of the most significant digit.
9—represents a numeric character 0 through 9 that is replaced by asterisks (***) in reports when the numeric value overflows the field.

A blank line follows the Statistics in the report.

6.2 Problems with Reports Using JOIN

If you erroneously specify events to be joined that do not share an item value, the Oracle Trace reporter completes successfully; however, the file does not contain a report. For the ATM_SAMPLE and the RDBVMS facility, the command in Example 6–1 will not produce a report.

Example 6–1 Example of Specifying Incorrect Items for a Join

```
$ COLLECT REPORT FORMATTED_SAMPLE.RDB -
        /TYPE=SUMMARY -
        /STATISTICS=ALL -
        /EVENTS=(BALANCE_EVENT,WITHDRAW_EVENT,TRANSACTION)-
        /JOIN=(CROSS_FAC_14) -
        /OUTPUT=FAILED_JOIN.RPT
```

The command in Example 6–2 produces the result in Section 3.2.1.

Example 6–2 Example of a Failed Join Report

```
25-SEP-1992 10:10                Summary Report                        Page 1
Selection: SAMPLE_SELECTION                                    Oracle Trace 2.2

Event:  Balance         In Facility:  ATM_SAMPLE      Version:  V2.2

%EPC-I-RPQU_NODATFND, No Data Found For Report
```

The command failed because all of these events did not share the same value for CROSS_FAC_14 as shown in the following table:

Event	Value for CROSS_FAC_14
WITHDRAW_EVENT	3
BALANCE_EVENT	2
TRANSACTION_EVENT	2,3

Although the WITHDRAW_EVENT and BALANCE_EVENT each share a common value for the CROSS_FAC_14 item with the TRANSACTION event (2,3), they do not share a common value between them by which Oracle Trace can join them.

The /JOIN qualifier connects events in a report based on a shared item value. If you want to use the /JOIN qualifier successfully, you must have a general understanding of how events come to share a value for an item, and a knowledge of the specific events that share item values in the facilities on which you are reporting.

If you are reporting on Oracle-supplied facilities such as Oracle Rdb, consult the documentation accompanying these facilities to learn which events share item values, or examine the facility definitions for these facilities. Be sure that the events that you want to join share an item value.

If you are reporting on an application that you have instrumented with Oracle Trace routines, you determine which events share which item values based on how you relate these events by using the RELATE keyword of the /OPTIONS qualifier of the CREATE DEFINITION command. You can also examine the facility definitions for these facilities to determine if the events that you want to join share the same item value.

7

Oracle Trace Formatting and Reporting Commands

Oracle Trace provides a command-line interface. To use the Oracle Trace commands, preface them with the keyword COLLECT. For example:

```
$ COLLECT SHOW VERSION
Oracle Trace Version V2.2
$
```

For better user interface performance, you can enter the Oracle Trace command environment by entering the COLLECT command with no arguments. This eliminates binding to the history and administration databases for each command. Oracle Trace prompts you for commands until you return to DCL command level with the EXIT command. For example:

```
$ COLLECT
Trace> SHOW VERSION
Oracle Trace Version V2.2
Trace> EXIT
$
```

This chapter describes the format and use of Oracle Trace formatting and reporting commands. Table 7–1 summarizes these commands.

Table 7–1 Oracle Trace Commands

Command	Description
@ (Execute Procedure)	Executes the commands in a command file as if you had typed them at the Oracle Trace prompt.
FORMAT	Formats one or more Oracle Trace data files into a formatted data file or database.
HELP	Displays requested information about the Oracle Trace commands.
REPORT[1]	Generates a report based on formatted data from one or more collections.

[1]Following the REPORT command section are separate reference sections that describe the /OPTION qualifier keywords: EVENT, ITEM, JOIN, and RESTRICTION.

@ (Execute Procedure)

The at sign (@) means execute, just as in DCL. When you type @ and the name of an indirect command file, Oracle Trace executes the statements in that file as if you had typed them one at a time at the Oracle Trace prompt. The command file must be an OpenVMS text file that contains Oracle Trace commands.

Format

@ file-spec

Parameter

file-spec
The name of the indirect command file. You can specify a full OpenVMS file specification, a file name, or a logical name. If you specify a file name, Oracle Trace looks in your current default OpenVMS directory for a file by that name. The file must contain valid Oracle Trace commands. The default file type is COM.

Description

This command is useful because it allows you to prepare a sequence of commands that you use often and that must be repeated identically each time you issue them. For example, if you generate weekly or monthly reports, you will want the same parameters defined each time.

Example

```
Trace> @DISK$USER1:[SMITH.COMS]SCHEDULE_A_COLLECTION.COM
```

Executes the commands in the file SCHEDULE_A_COLLECTION.COM, where the command file contains the following lines:

```
CREATE SELECTION VAX_INFO /OPTIONS
  FACILITY RDBVMS
  FACILITY ACMS/CLASS=ALL
  EXIT
SCHEDULE COLLECTION MY_TEST MY_DATA.DAT -
  /SELECTION=VAX_INFO -
  /BEGIN=11:00 /END=12:00
EXIT
```

FORMAT

Formats one or more Oracle Trace data files into one formatted data file or database.

Format

FORMAT data-file[, . . .] [formatted-file-or-database]

Command Qualifiers	Defaults
/[NO]CDDPLUS_DEFINITIONS=pathname	/NOCDDPLUS_DEFINITIONS
/ELAPSED_TIME	/NOELAPSED_TIME
/[NO]FILELIST	/NOFILELIST
/[NO]MERGE	/NOMERGE
/RDBVMS_OPTIMIZATION=(param=value[, . . .])	See text
/TYPE=output-type	/TYPE=RDBVMS

Although you can use the FORMAT command to produce both Oracle Rdb databases and RMS files, you can only use the CDDPLUS_DEFINITIONS, ELAPSED_TIME, and RDBVMS_OPTIMIZATION qualifiers when you produce an Oracle Rdb database.

Parameters

data-file
Must be either of the following:

- The name of a data capture file produced by data collection. You can also specify data capture files from more than one collection, provided all the collections reference the same facility selection.
- The name of a file that contains a list of data collection file names produced by one or more collections. (See the description of the /FILELIST qualifier.)

If the data collection files contain data from collections that are associated with different facility selections, Oracle Trace issues an error message and no formatting is done.

You cannot use any wildcard characters in the data file specifications. The default device and directory are your default OpenVMS device and directory. The default file type is DAT.

formatted-file-or-database
Specifies the name of the formatted data file or database that Oracle Trace creates to store the data. The default device and directory are your OpenVMS default device and directory. The /TYPE qualifier determines whether Oracle Trace creates a VAX RMS file or an Oracle Rdb database.

For VAX RMS formatted files, the default file type is DAT. For Oracle Rdb databases, you *cannot* specify a file type because Oracle Rdb creates both a database and a snapshot file. If you do specify a file type and request an Oracle Rdb database, Oracle Trace returns an error message and does not format the file.

The formats of both the VAX RMS file and the Oracle Rdb database are described in Appendix A.

Note

You can produce Oracle Trace reports only from a formatted Oracle Rdb database.

Qualifiers

/CDDPLUS_DEFINITIONS=pathname
/NOCDDPLUS_DEFINITIONS (default)
Specifies whether to create CDD/Repository record definitions for the formatted database or data file records. If you request CDD/Repository record definitions, you must specify a pathname where the definitions will be stored in the CDD/Repository dictionary.

You can only specify the /CDDPLUS_DEFINITIONS qualifier when you are formatting your data collection files into an Oracle Rdb formatted database. This qualifier is ignored if you generate VAX RMS formatted files.

/ELAPSED_TIME
NOELAPSED_TIME (default)
Calculates the elapsed time for each duration event in the data collection file and stores this information in the formatted Oracle Rdb database for use by report generators other than Oracle Trace. This qualifier is ignored if you are producing an RMS file (/TYPE=RMS).

You do not need this qualifier if you are producing Oracle Trace reports because Oracle Trace calculates elapsed time automatically; therefore, you only use this qualifier to include elapsed time in a formatted Oracle Rdb database from

which you are generating your own reports. Using this qualifier can increase formatting time.

/FILELIST
/NOFILELIST (default)
Specifies that the first parameter to the FORMAT command is a **file list**, which is a file containing a list of file specifications for the data files you want to format. For example:

```
$ COLLECT FORMAT MY_LIST.TXT /FILELIST MY_DATA
```

Each file specification must be on a separate line within the file list. For example, MY_LIST.TXT contains the following lines:

```
DISK$USER1:[SMITH.COLLECTOR]DATA1.DAT
DISK$USER2:[DATA]DATA2.DAT
DISK$USER3:[DATA]DATA3.DAT
```

The default file type for the file list is TXT.

/MERGE
/NOMERGE (default)
Specifies that the data in the data files should be merged with existing data in the specified formatted output file or database. The /MERGE qualifier lets you combine data from previous collections (that have already been formatted) with data from new collections. Only data from collections scheduled using the same facility selection can be combined.

When you specify the /MERGE qualifier and the output file or database does not exist, Oracle Trace issues an error and does not perform any formatting. If you do not specify /MERGE and the output file already exists, Oracle Trace creates a new version of the file using the same name.

/RDBVMS_OPTIMIZATION=(param=value[, . . .])
Specifies the use of optimization parameters for creating an Oracle Rdb formatted database. Table 7–2 shows the parameters and their default. Each parameter is described in the following section under Optimization Parameters.

Table 7–2 Oracle Rdb Optimization Parameters

Optimization Parameter	Default Value
ALLOCATION	2000 database pages
BUFFER_SIZE	30 blocks
[NO]JOURNAL	JOURNAL
MIN_EXTENT	500 pages
NUM_BUFFERS	30 buffers
[NO]STRING_OPTIMIZATION	STRING_OPTIMIZATION
[NO]VIEWS	NOVIEWS

/TYPE=output-type

Specifies the format of the formatted data file. Valid format types are:

RDBVMS
RMS

If you use the RDBVMS keyword, Oracle Trace creates an Oracle Rdb database that you can use with the REPORT command to generate Oracle Trace reports. If you use the RMS keyword, Oracle Trace creates a VAX RMS sequential file, and you cannot use Oracle Trace to produce reports. You will need to write your own reports. The choice of output types lets you select the format best suited for your purposes. With either type, you can choose to write your own reports by interpreting the data through user-written reporting programs or a report generator such as DEC DATATRIEVE.

RDBVMS is the default output type. The /TYPE=RMS qualifier is provided for users who plan to create their own reports based on the collected data. Appendix A describes the Oracle Rdb database and Appendix B describes the format of the VAX RMS file.

Optimization Parameters

ALLOCATION=number-of-pages

Specifies the number of database pages allocated for the data file produced when you create an Oracle Rdb database. You may want to increase the value of this qualifier when you create a very large Oracle Rdb database from your data collection files. The recommended value for the ALLOCATION parameter is based on the following formula:

$$number\text{-}of\text{-}pages = (4000 + size\text{-}of\text{-}DCF)/2$$

Where *size-of-DCF* is the size in blocks of all data collection files.

The default allocation is 2000 pages.

BUFFER_SIZE=number-of-blocks
Specifies the number of blocks allocated per buffer for use in creating an Oracle Rdb database. Valid buffer sizes range from 1 to 64.

The BUFFER_SIZE and NUM_BUFFERS parameters enable you to reduce the direct I/O required during formatting operations. However, the direct I/O is inversely proportional to the page faulting required; as you reduce the direct I/O, you increase the process' page faulting because the I/O operations have been transferred to the paging disk. In general, you can safely increase the number of buffers and the buffer size only if your system has a large amount of memory. You should experiment with different values to determine the optimal size and number of buffers for your application.

The default buffer size is 30 blocks.

[NO]JOURNAL
Specifies whether the database should be recoverable in the event of a formatting failure. If you specify NOJOURNAL, the database will be corrupted if formatting fails. The disadvantage of using JOURNAL, is that formatting operations take longer to complete.

Note that if you specify NOJOURNAL when formatting a group of data collection files and one of the files fails to format correctly, the database will be corrupted from that point on in the formatting. In this case, the format operation may fail completely. If you are formatting a group of data collection files for the first time, using NOJOURNAL provides little risk. If a failure occurs, you can format the files again. However, if you use NOJOURNAL to merge collection files with an existing database, Oracle Corporation recommends that you back up the database first.

The default is JOURNAL.

MIN_EXTENT=number-of-pages
Specifies the minimum number of pages used for the data file extent. You may want to increase the value of this qualifier when creating a very large Oracle Rdb database from your data collection files.

The default minimum extent is 500 pages.

NUM_BUFFERS=number-of-buffers
Specifies the number of buffers allocated for the creation of an Oracle Rdb database. The number is an unsigned integer greater than zero.

The BUFFER_SIZE and NUM_BUFFERS parameters enable you to reduce the direct I/O required during formatting operations. However, the direct I/O is inversely proportional to the page faulting required; as you reduce the direct I/O, you will increase the process' page faulting because the I/O operations have been transferred to the paging disk. In general, you can safely increase the number of buffers and the buffer size only if your system has a large amount of memory. You should experiment with different values to determine the optimal size and number of buffers for your application.

The default number of buffers is 30.

[NO]STRING_OPTIMIZATION
Specifies whether string storage optimization should be performed using a segmentation storage scheme. This parameter can greatly reduce the size of the formatted database for data of large ASCIC, ASCIW, and FIXED_ASCIC types. However, it will be more difficult to write report generators for a database using this optimization. Oracle Trace reporting is unaffected by the optimization.

The STRING_OPTIMIZATION parameter takes a list of the following parameters:

- FIRST_SEGMENT_SIZE=n-bytes—The size of the first string segment stored in the event data relation.
- SEGMENT_SIZE=n-bytes—The size of additional segments stored in the segmented string relation.

The default is STRING_OPTIMIZATION and the default values of the optimization parameters are based on the data specified in the facility definition.

Oracle Rdb has a limit on the size of a tuple (data row). This limit affects the formatting of data collection files with large string data. You should not specify a first segment size so large that the total event table tuple size exceeds the Oracle Rdb limit. For example, assume that the tuple limit is 32,000 bytes. If a data collection file has an event that contains three items of type ASCIW with a maximum size of 12,000 bytes each, the Oracle Trace formatting fails if the first segment size is 12,000. (12,000 bytes times 3 items is 36,000 bytes.)

For Oracle Rdb Version 3.1B-0, the tuple size limit is slightly less than 32,000 bytes. For Oracle Rdb Version 4.0, the tuple size limit is slightly less than 64,000 bytes.

[NO]VIEWS
Specifies whether views should be created for the event-data relations. If so, Oracle Trace creates four views for each event-data relation.

FORMAT

For data collection files with a large number of different event types, you can greatly increase formatting performance by using the NOVIEWS parameter. Note that Oracle Trace reporting does not need views.

The default is NOVIEWS.

Description

Oracle Trace collects raw data from the facilities that you select. Before you produce tabular reports from it, you must format the raw data into either an Oracle Rdb database or a VAX RMS file. To use Oracle Trace reporting, you must specify /TYPE=RDBVMS (or use the default to the /TYPE qualifier) to create an Oracle Rdb database. For example, the following command creates a formatted Oracle Rdb database named ALL_MY_DATA.RDB:

```
$ COLLECT FORMAT DUA0:[DATA]FOO1.DAT,DUA1:[DATA]FOO2.DAT ALL_MY_DATA
```

Examples

1. ```
   $ COLLECT FORMAT MY_COLLECTION.DAT TEST_DATA
   ```

   Formats the data in the file MY_COLLECTION.DAT and stores the formatted data in an Oracle Rdb database in your current default OpenVMS device and directory under the name TEST_DATA.RDB.

2. ```
   $ COLLECT FORMAT MY_LIST.TXT /FILELIST MY_DATA
   ```

   Formats the data in the files listed in MY_LIST.TXT and stores the formatted data in an Oracle Rdb database in your current default OpenVMS device and directory under the name MY_DATA.RDB. MY_ LIST.TXT contains a list of data file names, one per line.

3. ```
   $ COLLECT FORMAT WEEK3.DAT JUNE_DATA /MERGE
   ```

   Formats the data in WEEK3.DAT and merges it with the data already in the previously formatted JUNE_DATA.RDB database.

4. ```
   $ COLLECT FORMAT ONE_BIG_FILE.DAT ALL_DATA -
   _$ /RDBVMS_OPTIMIZATION=(NUM_BUFFERS=40, BUFFER_SIZE=50, NOVIEWS, -
   _$ STRING_OPTIMIZATION=(FIRST=32,SEGMENT=64), ALLOCATION=10000)
   ```

   Formats the data in ONE_BIG_FILE.DAT and stores the formatted data in an Oracle Rdb database named ALL_DATA.RDB. Because the file is very large, the normal formatting parameters are increased to provide more optimal performance.

HELP

Displays information about Oracle Trace and Oracle Trace commands.

Format

HELP [topic] . . .

Parameter

topic
Specifies the subject about which you want information from the HELP library. You can specify the names of one main topic and up to eight subtopics with the HELP command.

Example

```
Trace> HELP CREATE
  CREATE

    Creates a facility definition or a facility selection.

  Additional information available:

  DEFINITION     SELECTION

CREATE Subtopic?
```

Invokes Oracle Trace HELP to display information about the CREATE DEFINITION and CREATE SELECTION commands and prompts for further information.

REPORT

Generates a report based on formatted data from one or more collections.

Format

REPORT formatted-database

Command Qualifiers	Defaults
/BEFORE="time"	All data in file
/EVENTS=([facility.]event-name[, . . .])	/EVENTS=*
/FACILITY=(facility-name[, . . .])	/FACILITY=*
/GROUP_BY=([[facility.]event.]item[, . . .])	See text
/INTERVAL=time-unit	/INTERVAL=MINUTES
/ITEMS=([[facility.]event.]item-name[, . . .])	/ITEMS=*
/JOIN=([[facility.]event.]item= . . . ,[, . . .])	See text
/LENGTH=number-of-lines	/LENGTH=66
/[NO]OPTIONS=file-spec	/NOOPTIONS
/OUTPUT=file-spec	/OUTPUT=SYS$OUTPUT
/SINCE="time"	All data in file
/STATISTICS=(stat-type[, . . .])	/STATISTICS=COUNT
/TITLE="text-string"	/TITLE=""
/TYPE=report-type	/TYPE=SUMMARY
/WIDTH=number-of-columns	/WIDTH=80

Parameter

formatted-database
Specifies the name of a formatted database created with the FORMAT command. If the specified parameter is not an Oracle Rdb database, Oracle Trace issues an error message and generates a 0-block output file.

This is a required parameter.

Qualifiers

/BEFORE="dd-mmm-yyyy hh:mm:ss"
Specifies that Oracle Trace reports on all data collected at or before the indicated time. By default, Oracle Trace reports on all data in the formatted database.

If you use the /BEFORE qualifier, you must specify the time as an absolute date and time using a quoted string. You cannot use an OpenVMS delta time.

/EVENTS=([facility.]event-name[, . . .])
Specifies one or more events to report on. By default, Oracle Trace reports on the data collected for all facilities and events contained in the formatted database. You can use the /EVENTS qualifier to restrict the report to specific events. You must indicate the facility name if the event is not unique within the formatted data file.

The /EVENTS and /OPTIONS qualifiers are mutually exclusive and produce different kinds of reports. Using the /EVENTS qualifier on the REPORT command produces a report with consistent information for each of the events that you specify. All of the characteristics that you specify for the report pertain to all of the events. On the other hand, using the /OPTIONS qualifier allows you to specify unique characteristics for each event in a report.

/FACILITY=(facility-name[, . . .])
Specifies one or more facilities to report on. By default, Oracle Trace reports on the data collected for all facilities and events contained in the formatted database. You can use the /FACILITY qualifier to restrict the report to specific facilities.

If you specify the /FACILITY qualifier, you do not have to use the /EVENTS qualifier. If you do, the effects are cumulative: for example, if you specify two facilities and one event, Oracle Trace reports on that event for both facilities, provided both facilities contain the event. Note that an error occurs if you specify a facility that does not contain the specified events.

The /FACILITY and /OPTIONS qualifiers are mutually exclusive. Using the /FACILITY qualifier on the REPORT command produces a report with consistent information for the events in the facilities that you specify. All report characteristics that you specify for the report pertain to all of the events. On the other hand, using the /OPTIONS qualifier allows you to specify unique characteristics for each event in a report.

/GROUP_BY=([[facility.]event-name.]item-name[, . . .])
Specifies the items by which to group event occurrences. The /GROUP_BY qualifier pertains only to items whose value distinguishes event occurrences. For example, CLIENT_PC item in Oracle Rdb contains the value of the program counter, and it changes with each event occurrence. Therefore, specifying the CLIENT_PC item as a value to the /GROUP_BY qualifier produces a report in which event occurrences are grouped in ascending order based on the value of the CLIENT_PC item.

You can use the facility.event-name portion of the /GROUP_BY clause to specify unique items based on the data in the formatted database. However, when you specify a facility, you must also specify an event. If you use the facility name to identify an item, you must also specify the event name.

In Summary and Frequency reports, the value of the item you specify for the /GROUP_BY qualifier determines how the report groups event occurrences. Events with equivalent values for the item will be grouped together. The final group displays grand total statistics for the entire report.

Detail reports list event occurrences in ascending order based on the item values. If the event is a duration event, the end value of the item is used, providing the item was collected on the end event. Otherwise, the start value is used.

The /GROUP_BY qualifier is a way to display level and string items that Oracle Trace does not usually display in reports. For segmented ASCII strings, the /GROUP_BY qualifier is based on the first segment in the event relation. If two items are not equal, but the difference does not occur until after the first segment, the report considers them equal.

If you use the /GROUP_BY qualifier with the /JOIN qualifier, you must specify unique values for each qualifier. The report first orders events based on the value of the shared item, and within each group it orders events based on the value of the /GROUP_BY qualifier.

/INTERVAL=time-unit

Specifies the reporting interval for Frequency reports. An interval can be a second, minute, or an hour. For each interval within the collection period, the Frequency report displays a timestamp for the interval and the number of event occurrences within the interval. The Frequency report does not display intervals in which no event occurred.

You can use the SECONDS, MINUTES and HOURS keywords to specify an interval. The default interval is one minute.

The /INTERVAL qualifier is valid for Frequency reports only. Oracle Trace ignores the qualifier for other types of reports. Use the /TYPE qualifier to specify a Frequency report.

/ITEMS=([[facility.]event-name.]item-name[, . . .])
/NOITEMS

Specifies the item or items on which to report. If you specify a facility, you must also specify the event name. Using the /ITEMS qualifier without specifying a facility or an event name reports on every facility and event with which the item is associated.

The /ITEMS qualifier is valid for Detail and Summary reports only; it is ignored for the Frequency report. The /ITEMS qualifier is especially useful for limiting the size of Detail reports.

If you specify /NOITEMS on a Summary report, Oracle Trace uses /STATISTICS=COUNT and ignores any other statistics specified in the command.

/JOIN=([[facility.]event-name.]item-name=[[facility.]event-name.]item-name[, . . .])
Specifies the items by which to connect events in a report, based on a shared item value.

Note

Oracle Corporation recommends that you read the information in Chapter 5 before you use the /JOIN qualifier.

You can use the facility.event-name portion of the /JOIN clause to specify unique items based on the data in the formatted database. However, when you specify a facility, you must also specify an event. The items being joined must be the same data types and must be shared by the events as defined in the facility definition. The RELATE keyword of the /OPTIONS qualifier of the CREATE DEFINITION command relates events based on shared item values.

If you use both the /GROUP_BY and the /JOIN qualifiers in a single report, you must specify unique values for each of these qualifiers.

Joining items creates a hierarchy within a set of events. The facility.event.item portion of the join clause specifies the hierarchy. For example, if the event list contains events EVENT_A and EVENT_B, then the following command joins the events on ITEM_A for all facilities that contain either of the two events:

```
/EVENTS=(EVENT_A, EVENT_B)
/JOIN=(ITEM_A)
```

You can create multiple hierarchical levels by entering lists of items that are to be joined. Each list is delimited with a comma in the form: ITEMA=ITEMB,...,ITEMX=ITEMY.

For each event in the event list (/EVENTS=EVENT_A,EVENT_B,EVENT_C) the Reporter orders events based on the shared item values. The index page on the report shows the facility, event, and first join item value.

/LENGTH=number-of-lines
Specifies the length of each page of the report in number of lines. The minimum page length is 40 lines. The default page length is 66 lines.

/OPTIONS[=file-spec]
/NOOPTIONS (default)
Specifies a report that reports unique characteristics for each event. You can use the /OPTION qualifier in two ways:

- By typing /OPTIONS on the REPORT command line and entering keywords and keyword qualifiers interactively, in response to the OPTIONS> prompt.
- By creating a file containing keywords and keyword qualifiers and supplying the file name as a value to the /OPTIONS qualifier.

In both cases, you must specify each keyword on a separate line.

The /OPTIONS qualifier keywords are:

- EVENT—Denotes the beginning of a subreport and names the event to which it pertains. All subsequent keywords pertain to this event, until you specify the EVENT keyword again.
- ITEM—Allows you to override the display characteristics that are defined in the facility definition for items associated with this event.
- JOIN—Allows you to join events across subreports, based on a shared item value.
- RESTRICTION — Allows you to limit the event report to data that meets the criteria that you specify.

Descriptions of these keywords follow. See Chapter 4 for an example of a report produced using the /OPTIONS qualifier and keywords.

The /EVENTS and /FACILITY qualifiers are mutually exclusive with the /OPTIONS qualifier. Using the /FACILITY and /EVENTS qualifiers on the REPORT command produces a report with consistent information for each of the events in each of the facilities that you specify. All report characteristics that you specify for the report pertain to all of the events. On the other hand, using the /OPTIONS qualifier allows you to specify unique characteristics for events in a report.

The /OPTIONS qualifier is also useful when your Oracle Trace command is greater than 255 characters. Using the /OPTIONS qualifier lets you shorten the command by putting some of the command keywords and qualifiers in the options file.

/OUTPUT=file-spec
Specifies the destination for the report. By default, Oracle Trace displays the report on the SYS$OUTPUT device (usually, your terminal). The /OUTPUT qualifier can redirect the output to a file or another device.

/SINCE="dd-mmm-yyyy hh:mm:ss"
Specifies that Oracle Trace reports on all data collected at or after the specified time. By default, Oracle Trace reports on all the collected data in the formatted database.

If you use the /SINCE qualifier, you must specify the time as an absolute date and time using a quoted string. You cannot use an OpenVMS delta time.

/STATISTICS=(stat-type[, . . .])
Specifies the statistics to include for each data item in a Summary report. Valid statistics are:

ALL
COUNT (default)
MAXIMUM
MEAN
MINIMUM
STANDARD_DEVIATION
TOTAL
95_PERCENTILE

The /STATISTICS qualifier is valid for Summary reports only. Oracle Trace ignores the qualifier for Detail and Frequency reports.

/TITLE="text-string"
Specifies the title of the report. The title is displayed at the top of each page of the report. The title is a text string that can contain any printable characters. The string must be enclosed with quotation marks (" ").

/TYPE=report-type
Specifies the type of report to create. You can use the keywords SUMMARY, DETAIL, and FREQUENCY to specify the type of report that you want. The default report type is Summary. Table 7–3 describes the reports.

Table 7–3 Oracle Trace Reports

Report Name	Purpose	Description	Associated Qualifiers	See Section
Summary (default)	For evaluating event resource use for performance and capacity planning purposes.	Summarizes statistics about the collected data.	/ITEMS[1], /STATISTICS[2], /FACILITIES, /EVENTS, /GROUP_BY, or /OPTIONS	3.3
Detail	For debugging instrumented applications or application debugging.	Contains values of the items collected for each event.	/FACILITIES, /EVENTS, and /ITEMS[1] or /OPTIONS as described in Chapter 4.	3.5
Frequency	For tracing the execution of an instrumented application.	Summarizes event occurrences based on a selected interval of seconds, minutes, or hours.	/INTERVAL[3], /FACILITIES, /EVENTS, /GROUP_BY, or /OPTIONS	3.6

[1]The /ITEMS qualifier is valid only for the Detail and Summary reports.

[2]The /STATISTICS qualifier is valid only for Summary reports.

[3]The /INTERVAL qualifier is valid only for Frequency reports.

/WIDTH=number-of-columns

Specifies the width of each page of the report in number of characters. Valid widths are 80 and 132. If you specify a value other than 80 or 132, or if you do not specify a width, Oracle Trace uses 80 characters.

Description

In the simplest case, the REPORT command creates a Summary report, displayed to SYS$OUTPUT, on all of the data in the formatted database. For example, the following command generates a report based on the data in the formatted database MY_TEST.RDB:

```
$ COLLECT REPORT MY_TEST.RDB
```

You can further refine the report with the /FACILITY and /EVENTS qualifiers. For example, the following command reports on only the ACMS PROCEDURE_CALL events:

```
$ COLLECT REPORT MY_TEST -
_$ /FACILITY=ACMS /EVENT=PROCEDURE_CALL
```

The /BEFORE, /INTERVAL, /SINCE, and /STATISTICS qualifiers provide additional customizing features that you may want to use.

You can specify all of the characteristics of the report using the REPORT qualifiers listed in the previous paragraph. However, the qualifiers are global; that is, the statistics that you request with the /STATISTICS qualifier apply to all of the events that you ask to be included in the report. You can override the qualifiers for each event. If you want to define separate characteristics for each event, you can use the /OPTIONS qualifier.

See Chapter 4 for an example of the report characteristics that you can specify in the report options file. Also, following the REPORT command in this reference section are descriptions of the /OPTIONS qualifier keywords : EVENT, ITEM, JOIN, and RESTRICTION.

Examples

1.
```
$ COLLECT REPORT MY_REPORT /OUTPUT=MY_REPORT.TXT
```

   Creates a Summary report on all of the data in the formatted file MY_REPORT.RDB and stores the report in the file MY_REPORT.TXT.

2.
```
$ COLLECT REPORT MY_REPORT /OUTPUT=MY_REPORT.TXT -
_$ /FACILITY=ACMS /EVENT=TASK /STATISTICS=(MINIMUM,MAXIMUM,MEAN) -
_$ /SINCE="16-SEP-1991 16:27:00"
```

   Creates a Summary report of the ACMS task data stored in the formatted database MY_REPORT.RDB after a specific time. The report lists the minimum, maximum, and mean values of task data items. Oracle Trace stores the report in the file MY_REPORT.TXT.

3.
```
$ COLLECT REPORT MY_REPORT /OUTPUT=MY_REPORT.TXT /OPTIONS
Options>EVENT TASK /FACILITY=ACMS /STATISTICS=MAXIMUM -
_Options>/GROUP_BY=TASK_NAME
_Options>CTRL/Z
```

   Creates a Summary report of ACMS task data from the formatted database MY_REPORT.RDB. The report lists the maximum number of occurrences of the TASK event, grouped by task name. Oracle Trace stores the report in the file MY_REPORT.TXT.

REPORT Options—EVENT

Denotes the beginning of an event-specific subreport, defines the report's characteristics, and names the event to which it pertains. Subsequent /OPTION keywords that you specify pertain to this event subreport until you specify another event. This is a keyword to the /OPTIONS qualifier of the REPORT command.

Note that the qualifiers to the EVENT keyword perform the same function as the REPORT command qualifiers /FACILITY, /EVENT, /ITEMS, or /GROUP_BY, to which they correspond. Table 7–4 shows how the REPORT command qualifiers and corresponding /OPTION keyword qualifiers interact with one another.

Table 7–4 Interaction Between REPORT Qualifiers and Their Corresponding /OPTIONS Keyword Qualifiers

When the REPORT qualifier[1] is . . .	and the /OPTIONS keyword qualifier is . . .	Then . . .
specified	specified	the /OPTIONS keyword overrides the corresponding REPORT qualifier for the subreport.
not specified	specified	the /OPTIONS keyword qualifier determines the characteristic of the subreport.
specified	not specified	the REPORT qualifier determines the characteristics of the subreport.
not specified	not specified	the default determines the characteristics of the subreport.

[1]For example /FACILITY, /EVENT, /ITEMS/ and /GROUP_BY

Format

EVENT event-name

Command Qualifiers	Defaults
/BEFORE=time	All data in file
/FACILITY=facility-name	None
/GROUP_BY=([[facility.]event-name.]item-name[, . . .])	See text
/INTERVAL=time	See text

/[NO]ITEMS=([[facility.]event-name.]item-name[, . . .])	/ITEMS=*
/SINCE=time	All data in file
/STATISTICS=(stat-type[, . . .])	See text
/[NO]SUBTITLE="text-string"	/NOSUBTITLE
/TYPE=report-type	See text

Parameter

event-name
Specifies the name of an event to report on.

Qualifiers

/BEFORE="dd-mmm-yyyy hh:mm:ss"
Specifies that Oracle Trace reports on all data collected at or before the indicated time. By default, Oracle Trace reports on all data in the formatted database for this event.

If you use the /BEFORE qualifier, you must specify the time as an absolute date and time using a quoted string. You cannot use an OpenVMS delta time.

/FACILITY=facility-name
Specifies the name of the facility that contains the event.

/GROUP_BY=([[facility.]event-name.]item-name[, . . .])
Specifies the item or items by which to group event occurrences. In Summary and Frequency reports, the report statistics are divided into groups based on equivalent values of the specified items. The final group displays grand total statistics for the entire report.

In Detail reports, event occurrences are listed in ascending order based on the item values. If the event is a duration event, the end value of the item is used, providing the item was collected on the end event. Otherwise, the start value is used.

For segmented ASCII strings, the /GROUP_BY qualifier is based on the first segment in the event relation. If two items are not equal, but the difference does not occur until after the first segment, the report considers them equal.

/INTERVAL=time-unit
Specifies the reporting interval for a Frequency report pertaining to an event. An interval can be a second, minute, or hour. For each interval within the collection period, the Frequency report displays a timestamp for the interval and the number of event occurrences within the interval. The Frequency report does not display intervals in which no event occurred. Use the keywords

SECONDS, MINUTES, and HOURS to specify the interval you want. The default interval is one minute.

If an /INTERVAL qualifier is specified on the REPORT command line, the command line qualifier specifies the default interval. The /INTERVAL qualifier on an EVENT option overrides the default interval for this event only.

The /INTERVAL qualifier is valid for Frequency reports only. The qualifier is ignored for other types of reports.

/ITEMS=([[facility.]event.]item-name[, . . .])
/NOITEMS
Allows you to limit the subreport to items that you specify or to eliminate items from the subreport entirely. If you do not use this qualifier, the subreport will display statistics for all of the items associated with the event.

The /ITEMS qualifier is valid for Detail and Summary reports only. It is ignored for Frequency reports. If you specify /NOITEMS on a Summary report, Oracle Trace uses /STATISTICS=COUNT and ignores any other statistic that you specify in the command.

Do not specify /NOITEMS and /GROUP_BY on a Detail report. If you do, Oracle Trace ignores the /GROUP_BY qualifier and produces a report that contains only the report header information.

/SINCE="dd-mmm-yyyy hh:mm:ss"
Specifies that Oracle Trace reports on all data collected for this event at or after the specified time. By default, Oracle Trace reports on all the collected data in the formatted database for this event name.

If you use the /SINCE qualifier, you must specify the time as an absolute date and time using a quoted string. You cannot use an OpenVMS delta time.

/STATISTICS=(stat-type[, . . .])
Specifies the statistics to include for each item in a Summary subreport. You can use the following keywords to specify the statistics that you want:

95_PERCENTILE
ALL
COUNT (default)
MAXIMUM
MEAN
MINIMUM
STANDARD_DEVIATION
TOTAL

The /STATISTICS qualifier is valid for Summary subreports only. Oracle Trace ignores the qualifier for Detail and Frequency reports.

/SUBTITLE="text-string"
/NOSUBTITLE (default)
Specifies a subtitle for the report. The subtitle is displayed only at the top of the first page of the report. The subtitle is a text string that can contain any printable characters. The string must be enclosed with quotation marks (" ")

/TYPE=report-type
Specifies the type of subreport to create. You can use the DETAIL, FREQUENCY and SUMMARY keywords to specify the type of report that you want. The default report type is Summary.

Note

You cannot produce a report with more than one report type for the same event. If you want different reports for the same event, you must use two REPORT commands.

Description

You can specify keywords and keyword qualifiers to the /OPTION qualifier in either an options file or interactively at the Options> prompt. In either case, you must use the /OPTIONS qualifier to the REPORT command to specify options. Specify each option on a separate line. You can specify each option more than once.

REPORT Options—ITEM

Allows you to override the display characteristics that are defined in the facility definition by specifying new characteristics that pertain only to this event subreport. You specify this keyword using the /OPTIONS qualifier to the REPORT command.

The display characteristics that you can change are:

- Report heading
- Display width

Format

ITEM item-name

Command Qualifiers	Defaults
/REPORT_HEADER=text	See text
/WIDTH=number-of-columns	See text

Parameter

item-name
The name of the item for which to define the characteristics.

Qualifiers

/REPORT_HEADER=text
Specifies the text to display as a heading in the report when listing data for the specified item.

The facility definition specifies the default report heading.

/WIDTH=number-of-columns
Specifies the number of columns to reserve for displaying data for the specified item.

The facility definition specifies the default width.

REPORT Options—JOIN

Connects events with a shared item value across subreports. You specify this command using the /OPTIONS qualifier to the REPORT command.

Note

Oracle Corporation recommends that you read Chapter 5 before you use the JOIN keyword.

Format

JOIN item-list

Parameter

item-list
Specifies the items to connect in the report. The items may be prefixed by event or event and facility to make them unique based on the data in the formatted database. If you specify a facility, then you must also specify an event. The items being joined must be the same data type.

If you use the /GROUP_BY qualifier with the JOIN keyword, you must specify unique values for both.

The facility.event.item portion of the join clause specifies the hierarchy of events in which the report displays events. For example, if the /OPTIONS qualifier specifies EVENT EVENT_A and EVENT EVENT_B, then the following command joins the events on ITEM_A for all facilities that contain either of the two events:

```
JOIN item_a
```

You can create multiple hierarchical levels by simply entering lists of items that are to be joined. Each list is delimited with a comma in the form: ITEMA=ITEMB,...,ITEMX=ITEMY.

The reporter generates subreports ordered by the JOIN items for each event in the event list. The index page on the report shows the facility, event, and first join item value.

REPORT Options—RESTRICTION

Defines a subset of the data to report on in the subreport. You specify this as a keyword to the /OPTIONS qualifier of the REPORT command.

You can selectively report on data based on the following items:

- Collection names
- Image names
- Item values
- Node names
- Process IDs (EPID)

Format

RESTRICTION { COLLECTION collection-name [, . . .] | EPID process-ID [, . . .] | IMAGE image-name [, . . .] | ITEM item-name item-value | NODE node-name [, . . .] }

Parameters

collection-name
The name of one or more collections to use to select data for reporting. Only data collected during the specified collections is reported.

image-name
The name of the executable image that registered with the Oracle Trace Registrar process. Only data collected from processes running the specified image is reported.

The image names that you specify are used as wildcards in searching through the formatted database for the desired images. The effect is that the restriction is done on **image-name**. Any images that contain the image name that you specify are included in the subreport. Therefore, if you want to limit the subreport to a specific image, you must specify enough of the image name to distinguish it from images that might be a superset of the image name.

item-name
The name of an item to use to select data for reporting.

item-value
The value of the item to use to select data. Only records that match the item name and value are reported.

node-name
The name of one or more network nodes to use to select data for reporting. Only data collected on the specified nodes is reported.

process-ID
The EPIDs of one or more processes to use to select data for reporting. Only data collected from the specified processes is reported.

A

Oracle Rdb Database Format

This appendix describes the Oracle Rdb database file created by Oracle Trace when you specify the /TYPE=RDBVMS qualifier to the FORMAT command. Oracle Trace creates an Oracle Rdb database as the default if you do not specify the /TYPE qualifier.

You can use the Oracle Trace formatter to merge, format, and store the event data collected from multiple collections that reference the same facility selection into an Oracle Rdb database.

Only collections of identically defined selections can be merged into a single database. Identically defined selections have identical data for fields except those listed in Table A–1.

Table A–1 Fields Not Identically Defined

Relation	Records Not Identically Defined	Associated Fields Not Identically Defined
EPC$SELECTION	SELECTION_RECORD_ ID	SELECTION_NAME SELECTION_COMMENT
EPC$FACILITY	SELECTION_RECORD_ ID	FACILITY_NUMBER FACILITY_NAME FACILITY_VERSION FACILITY_DEFINITION_TIME COLLECTION_CLASS
EPC$EVENT	SELECTION_RECORD_ ID RELATION_NAME	FACILITY_NUMBER EVENT_NUMBER EVENT_NAME EVENT_HEADER EVENT_TYPE ITEM_FLAGS

(continued on next page)

Table A–1 (Cont.) Fields Not Identically Defined

Relation	Records Not Identically Defined	Associated Fields Not Identically Defined
EPC$ITEM	SELECTION_RECORD_ID	FACILITY_NUMBER ITEM_NUMBER ITEM_NAME ITEM_HEADER ITEM_TYPE ITEM_WIDTH ITEM_MAX_SIZE ITEM_USAGE ITEM_CHARACTERISTICS
EPC$EVENT_ITEM	SELECTION_RECORD_ID	FACILITY_NUMBER EVENT_NUMBER ITEM_NUMBER ITEM_POSITION ITEM_USAGE_POINT ITEM_USAGE_START ITEM_USAGE_END

A.1 The Data Types

Table A–2 shows the data types that Oracle Rdb uses to represent the Oracle Trace-supported data types.

Table A–2 Oracle Rdb Representations of Oracle Trace-Supported Data Types

Oracle Trace-Supported Data Type	Oracle Rdb Data Type
BYTE	SMALLINT
WORD	SMALLINT
LONGWORD	INTEGER
QUADWORD	QUADWORD
ASCIC	VARCHAR of up to 255 characters
ASCIW	VARCHAR of up to 65,533 characters
FIXED_ASCIC	VARCHAR of up to 255 characters

The database contains two types of relations that store two types of data.

1. Control information—collection and facility information

2. Collected event data

The following sections describe in detail the relations in the database.

Note

Some of the relations described here may be views that are defined over other base relations. The format of these base relations may change over time for improvements in performance or space utilization of the database. The names and formats of the relations or views and fields described here, however, are unlikely to change.

A.2 String Data Segmentation

Oracle Trace uses a form of string segmentation in the Oracle Rdb formatted database that improves formatting performance and saves storage space. This string segmentation applies to all data items of type FIXED_ASCIC, ASCIC, or ASCIW in all event data relations.

Oracle Trace stores all collected string data of types FIXED_ASCIC, ASCIC, or ASCIW in multiple segments. It stores the first segment of a string datum in the base event data relation, and additional segments in a separate relation for segmented strings only. The event relation contains one segmented-string relation for each base event data relation that has at least one data item of type FIXED_ASCIC, ASCIC, or ASCIW. A storage space of size zero for the first segment causes all string segments for this data field to be stored in the segmented-string relation. Oracle Trace stores strings that are small enough to fit in the first segment.

Within each event data relation, a string ID relates all segments of a single datum. The string ID is unique, and it resides in an event data relation field called <item_name>_STR_ID. For example, for point event data for item WHERE_BLOCK, the string ID is WHERE_BLOCK_STR_ID. For the end data of a duration event, the string ID is WHERE_BLOCK_END_STR_ID. A value of 0 (zero) in this field indicates that there is only one string segment: the first, for a datum.

EPC$1_40_REQUEST is an example of an event data relation where two of the data item, ARGUMENT_LIST and WHERE_BLOCK, are of the ASCIW type, and one item, TARGET_DBKEY, is of the ASCIC type. Therefore, three additional fields, with a domain of STR_ID_DOMAIN (type INTEGER), are also created for the string IDs of these data fields. Table A–3 describes the fields in these relations.

Table A–3 The Point Event Relation EPC$1_40_REQUEST

Field name	Oracle Rdb Data Type
COMP_STATUS	INTEGER
CODE_QUADWORD	QUADWORD
ARGUMENT_LIST	VARCHAR(x)
ARGUMENT_LIST_STR_ID	INTEGER
WHERE_BLOCK	VARCHAR(y)
WHERE_BLOCK_STR_ID	INTEGER
TARGET_DBKEY	VARCHAR(z)
TARGET_DBKEY_STR_ID	INTEGER

Segmented-string relations are named <event_data_relation_name>_ST, and have three fields. Table A–4 describes the fields in these relations.

Table A–4 A Relation for Segmented Strings

Field name	Oracle Rdb Data Type	Description
STR_ID	INTEGER	The string ID of the datum.
SEGMENT_NUMBER	SMALLINT	Segment sequence number, starts with 1.
STR_SEGMENT	VARCHAR(n)	String segment.

One index is created for each of these relations on the STR_ID field. The index is named the same as the relation itself.

Two data fields in two of the Oracle Trace system relations indicate the segment sizes of the segmented strings:

1. The ITEM_FIRST_SEGMENT_SIZE field, of domain ITEM_FIRST_ SEGMENT_SIZE_DOMAIN, a SMALLINT, in the EPC$EVENT_ITEM relation, indicates the storage size for the first string segment of the data item in the base event data relation.
2. SEGMENT_SIZE, of domain EVENT_SEGMENT_SIZE_DOMAIN, a SMALLINT, in the EPC$EVENT relation, indicates the segment size of all the additional string segments in the segmented-string relation for the event.

A.3 Relations for the Control Information

There are 14 relations that hold control information:

EPC$IDENT	EPC$FACILITY	EPC$IMAGE
EPC$SELECTION	EPC$EVENT	EPC$DCF_IMAGE
EPC$COLLECTION	EPC$ITEM	EPC$FACILITY_IMAGE
EPC$DCF	EPC$EVENT_ITEM	EPC$RELATIONSHIP_TABLE_ NAMES
EPC$REG_ PROCESS	EPC$PROCESS	

A.3.1 The EPC$IDENT Relation

The EPC$IDENT relation holds information of the Oracle Trace collector that collected the data.

The IDENT_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. Table A–5 describes the fields in the EPC$IDENT relation.

Table A–5 EPC$IDENT Relation

Field Name	Data Type/Size	Description
IDENT_RECORD_ID	Signed word	The EPC$IDENT record identifier.
PRODUCT_VERSION	Varying string, 10 bytes	The Oracle Trace product version string.
DCF_VERSION	Varying string, 10 bytes	The data collection file version string.
FORMAT_VERSION	Varying string, 10 bytes	The formatted database version string.

Bold signifies key fields.

A.3.2 The EPC$SELECTION Relation

The EPC$SELECTION relation holds the facility selection definition information.

The SELECTION_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. Table A–6 describes the fields in the EPC$SELECTION relation.

Table A–6 EPC$SELECTION Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier. Range: 1 to 999 inclusive.
SELECTION_NAME	Varying string, 32 bytes	The selection name.
SELECTION_COMMENT	Varying string, 80 bytes	The comment for the selection.

Bold signifies key fields.

A.3.3 The EPC$COLLECTION Relation

The EPC$COLLECTION relation holds the collection information.

The COLLECTION_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. The SELECTION_RECORD_ID field serves as a foreign key from relation EPC$SELECTION. Table A–7 describes the fields in the EPC$COLLECTION relation.

Table A–7 EPC$COLLECTION Relation

Field Name	Data Type/Size	Description
COLLECTION_RECORD_ID	Signed word	The EPC$COLLECTION record identifier.
COLLECTION_NAME	Varying string, 32 bytes	The collection name.
START_TIME	Date	The collection start time.

Bold signifies key fields.
Italic signifies foreign key fields.

(continued on next page)

Table A–7 (Cont.) EPC$COLLECTION Relation

Field Name	Data Type/Size	Description
END_TIME	Date	The collection end time.
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.

Bold signifies key fields.
Italic signifies foreign key fields.

A.3.4 The EPC$DCF Relation

The EPC$DCF relation holds the data collection file information.

The DCF_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. The COLLECTION_RECORD_ID field serves as a foreign key from relation EPC$COLLECTION. Table A–8 describes the fields in the EPC$DCF relation.

Table A–8 EPC$DCF Relation

Field Name	Data Type/Size	Description
DCF_RECORD_ID	Signed word	The EPC$DCF record identifier.
SYSTEM_ID	Integer, 6 bytes	The system identification of the node, obtain with an SYS$GETSYI call, specifying SYI$_NODE_SYSTEMID.
ACTIVATION_TIME	Date	Collection activation time.
TOTAL_FILES	Signed word	The total number of DCFs from the collection.
FILE_NUMBER	Signed word	The file number of this DCF.
FILE_NAME	Varying string, 255 bytes	The file name of this DCF.
COLLECTION_RECORD_ID	Signed word	The EPC$COLLECTION record identifier.

Bold signifies key fields.
Italic signifies foreign key fields.

A.3.5 The EPC$REG Relation

The EPC$REG relation holds the registration identification information.

The COLLECTION_RECORD_ID field serves as a foreign key from relation EPC$COLLECTION.

Table A–9 describes the fields in the EPC$REG relation.

Table A–9 EPC$REG Relation

Field Name	Data Type/Size	Description
COLLECTION_RECORD_ID	Signed word	The EPC$COLLECTION record identifier.
REG_ID	Varying string, 255 bytes	The registration ID.

Italic signifies foreign key fields.

A.3.6 The EPC$FACILITY Relation

The EPC$FACILITY relation holds the facility definition information.

The SELECTION_RECORD_ID and FACILITY_NUMBER together form the key field and are indexed. The SELECTION_RECORD_ID field also serves as a foreign key from relation EPC$SELECTION. Table A–10 describes the fields in the EPC$FACILITY relation.

Table A–10 EPC$FACILITY Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.
FACILITY_NUMBER	Signed word	The facility number.
FACILITY_NAME	Varying string, 27 bytes	The facility name.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

(continued on next page)

Table A–10 (Cont.) EPC$FACILITY Relation

Field Name	Data Type/Size	Description
FACILITY_VERSION	Varying string, 10 bytes	The facility version.
FACILITY_DEFINITION_TIME	Date	Facility definition creation date.
COLLECTION_CLASS	Varying string, 32 bytes	The collection class name.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

A.3.7 The EPC$EVENT Relation

The EPC$EVENT relation holds the event information for a facility.

The SELECTION_RECORD_ID, FACILITY_NUMBER, and EVENT_ NUMBER together form the key field for this relation and are indexed. The SELECTION_RECORD_ID field also serves as a foreign key from relation EPC$SELECTION. The FACILITY_NUMBER field also serves as a foreign key from relation EPC$FACILITY. Table A–11 describes the fields in the EPC$EVENT Relation.

Table A–11 EPC$EVENT Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.
FACILITY_NUMBER	Signed word	The facility each event is associated with.
EVENT_NUMBER	Signed word	The event number for this event.
EVENT_NAME	Varying string, 15 bytes	Name describing the event.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

(continued on next page)

Table A–11 (Cont.) EPC$EVENT Relation

Field Name	Data Type/Size	Description
EVENT_HEADER	Varying string, 15 bytes	The header used in reports.
EVENT_TYPE	Signed word	The event type. Possible values follow: • EPC$K_EVENT_TYPE_POINT(81)—Indicates a point event type. • EPC$K_EVENT_TYPE_DURATION(84)—Indicates a duration event type.
ITEM_FLAGS	Bit array, 256 bytes	The item flags for the event.
SEGMENT_SIZE	Signed word	Size of the segmented string.
RELATION_NAME	Varying string, 28 bytes	The name of the relation that holds the captured data for this event.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

A.3.8 The EPC$ITEM Relation

The EPC$ITEM relation holds the item information for a facility.

The SELECTION_RECORD_ID, FACILITY_NUMBER, and ITEM_NUMBER together form the key field for this relation and are indexed. The SELECTION_RECORD_ID field also serves as a foreign key from relation EPC$SELECTION. The FACILITY_NUMBER field also serves as a foreign key from relation EPC$FACILITY. Table A–12 describes the fields in the EPC$ITEM relation.

Table A–12 EPC$ITEM Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.
FACILITY_NUMBER	Signed word	The facility each item is associated with.
ITEM_NUMBER	Signed word	The item number for this event.
ITEM_NAME	Varying string, 15 bytes	Name describing the item.
ITEM_HEADER	Varying string, 15 bytes	The column header used in reports.
ITEM_TYPE	Signed word	The data type of the item.
ITEM_RADIX	Signed word	The radix of the item.
ITEM_WIDTH	Signed word	The column width used in reports.
ITEM_MAX_SIZE	Signed longword	The maximum size of the item.
ITEM_USAGE	Signed word	Usage of the item. Possible values follow: • EPC$K_ITM_U_LEVEL(1)—Indicates the item is a level. A level is a meter or gauge that indicates the "CURRENT" value of some metric. Its value may go up & down. • EPC$K_ITM_U_COUNTER(2)—Indicates the item is a counter. A counter indicates the number of times something occurred. Its value increases over time. • EPC$K_ITM_U_PERCENT(3)—Indicates the item is a percentage. A percentage is a type of level. • EPC$K_ITM_U_TEXT(4)—Indicates the item is text. • EPC$K_ITM_U_PRIVATE(5)—Indicates the item is for private use by the facility.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

(continued on next page)

Table A–12 (Cont.) EPC$ITEM Relation

Field Name	Data Type/Size	Description
ITEM_CHARACTERISTICS	Signed longword	Item value characteristics. Possible values follow: • EPC$K_ITM_CHR_PRINT(1)—Printable item value. • EPC$K_ITM_CHR_NONPRINT(2)—Non-printable item value.

Bold signifies key fields.
Bolded italic signifies key fields that are also foreign key fields.

A.3.9 The EPC$EVENT_ITEM Record

The EPC$EVENT_ITEM relation holds the relationship information between the events and items of a facility.

The SELECTION_RECORD_ID, FACILITY_NUMBER, EVENT_NUMBER, and ITEM_NUMBER together form the key field for this relation and are indexed. The SELECTION_RECORD_ID field also serves as a foreign key from relation EPC$SELECTION. The FACILITY_NUMBER field also serves as a foreign key from relation EPC$FACILITY. The EVENT_NUMBER field also serves as a foreign key from relation EPC$EVENT. The ITEM_NUMBER field also serves as a foreign key from relation EPC$ITEM. Table A–13 describes the fields in the EPC$EVENT_ITEM relation.

Table A–13 EPC$EVENT_ITEM Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.
FACILITY_NUMBER	Signed word	The facility each event-item pair is associated with.

Bolded italic signifies key fields that are also foreign key fields.

(continued on next page)

Table A–13 (Cont.) EPC$EVENT_ITEM Relation

Field Name	Data Type/Size	Description
EVENT_NUMBER	Signed word	The event number of the event-item pair.
ITEM_NUMBER	Signed word	The item number of the event-item pair.
ITEM_POSITION	Signed word	The item position in the event record.
ITEM_FIRST_SEGMENT_SIZE	Signed word	The size of the first segment of the segmented string.
ITEM_USAGE_START	Signed word	A value of one indicates that this data item will be collected in a start event, for the previous event number. A value of zero indicates otherwise.
ITEM_USAGE_END	Signed word	A value of one indicates that this data item will be collected in an end event, for the previous event number. A value of zero indicates otherwise.
ITEM_USAGE_POINT	Signed word	A value of one indicates that this data item will be collected in a point event, for the previous event number. A value of zero indicates otherwise.

Bolded italic signifies key fields that are also foreign key fields.

A.3.10 The EPC$PROCESS Relation

The EPC$PROCESS relation holds the process information.

The PROCESS_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. The NODE field is also indexed. The IDENT_ RECORD_ID field serves as a foreign key from relation EPC$IDENT. Table A–14 describes the fields in the EPC$PROCESS relation.

Table A–14 EPC$PROCESS Relation

Field Name	Data Type/Size	Description
PROCESS_RECORD_ID	Signed word	The EPC$PROCESS record identifier.
EPID	Signed longword	The extended process identity number.
USERNAME	Varying string, 12 bytes	The user name of the process.
ACCOUNT	Varying string, 8 bytes	The process account name.
UIC	Signed longword	The process user identification.
PROCESS_NAME	Varying string, 15 bytes	The process name.
CREATION_TIME	Date	Process creation time.
BASE_PRIORITY	Signed longword	The base priority of the process.
PROCESS_MODE	Signed longword	The process mode.
SYSTEM_ID	Integer, 6 bytes	The system identification of the node, obtained with a SYS$GETSYI call, specifying SYI$_NODE_SYSTEMID.
NODE_NAME	Varying string, up to 404 bytes	The full DECnet node name, obtained with a SYS$GETSYI call, specifying SYI$_NODENAME.
NODE	Varying string, up to 64 bytes	The abbreviated version of the previous DECnet node name.
CPU_TYPE	Varying string, 31 bytes	The CPU type, obtained with a SYS$GETSYI call, specifying SYI$_HW_NAME.
VMS_VERSION	Varying string, 8 bytes	The OpenVMS version, obtained with a SYS$GETSYI call, specifying SYI$_VERSION.
BOOT_TIME	Date	The boot time of the node, result of a SYS$GETSYI call when specifying SYI$_BOOTTIME.

Bold signifies key fields.
Italic signifies foreign key fields.

(continued on next page)

Table A–14 (Cont.) EPC$PROCESS Relation

Field Name	Data Type/Size	Description
IDENT_RECORD_ID	Signed word	The EPC$IDENT record identifier.

Bold signifies key fields.
Italic signifies foreign key fields.

A.3.11 The EPC$IMAGE Relation

The EPC$IMAGE relation holds the image information.

The IMAGE_RECORD_ID field is a formatter-introduced field used as the key field and is indexed. The PROCESS_RECORD_ID field serves as a foreign key from relation EPC$PROCESS. Table A–15 describes the fields in the EPC$IMAGE relation.

Table A–15 EPC$IMAGE Relation

Field Name	Data Type/Size	Description
IMAGE_RECORD_ID	Signed longword	The EPC$IMAGE record identifier.
IMAGE_NAME	Varying string, 255 bytes	The full file specification of the image.
ACTIVATION_TIME	Date	Image activation time.
LINK_TIME	Date	Image link time.
IMAGE_ID	Varying string, 15 bytes	The image identity number.
PROCESS_RECORD_ID	Signed word	The EPC$PROCESS record identifier.

Bold signifies key fields.
Italic signifies foreign key fields.

A.3.12 The EPC$DCF_IMAGE Relation

The EPC$DCF_IMAGE relation captures the relationship information between data collection files and execution images.

The DCF_RECORD_ID and IMAGE_RECORD_ID together form the key field for this relation and are indexed. The DCF_RECORD_ID field also serves as a foreign key from relation EPC$DCF. The IMAGE_RECORD_ID field also serves as a foreign key from relation EPC$IMAGE. Table A–16 describes the fields in the EPC$DCF_IMAGE relation.

Table A–16 EPC$DCF_IMAGE Relation

Field Name	Data Type/Size	Description
DCF_RECORD_ID	Signed word	The EPC$DCF record identifier.
IMAGE_RECORD_ID	Signed longword	The EPC$IMAGE record identifier.

Bolded italic signifies key fields that are also foreign key fields.

A.3.13 The EPC$FACILITY_IMAGE Relation

The EPC$FACILITY_IMAGE relation holds the relationship information between facilities and execution images.

The SELECTION_RECORD_ID, FACILITY_NUMBER, and IMAGE_RECORD_ID together form the key field for this relation and are indexed. The SELECTION_RECORD_ID field also serves as a foreign key from relation EPC$SELECTION. The FACILITY_NUMBER field also serves as a foreign key from relation EPC$FACILITY. The IMAGE_RECORD_ID field also serves as a foreign key from relation EPC$IMAGE. Table A–17 describes the fields in the EPC$FACILITY_IMAGE relation.

Table A–17 EPC$FACILITY_IMAGE Relation

Field Name	Data Type/Size	Description
SELECTION_RECORD_ID	Signed word	The EPC$SELECTION record identifier.

Bolded italic signifies key fields that are also foreign key fields.

(continued on next page)

Table A–17 (Cont.) EPC$FACILITY_IMAGE Relation

Field Name	Data Type/Size	Description
FACILITY_NUMBER	Signed word	The facility number.
FACILITY_VERSION	Varying string, 10 bytes	The facility version.
IMAGE_RECORD_ID	Signed longword	The EPC$IMAGE record identifier.
REG_ID	Varying string, 255 bytes	The facility registration ID.

Bolded italic signifies key fields that are also foreign key fields.

A.3.14 The EPC$RELATIONSHIP_TABLE_NAMES Relation

A tuple for each relationship is stored in the EPC$RELATIONSHIP_TABLE_NAMES table. This serves as an index (directory) to all relationship tables that are stored within the formatted database. The following table describes the field in the relationship table:

Field Name	Data Type/Size	Description
TABLE_NAME	Varying string, 31 bytes	Name of the table where the relationship resides.

Bold signify key fields.

The name of the table stored in the formatted database is explicitly indicated on the RELATE option of the CREATE DEFINITION command. Table A–18 describes the fields in the relation table.

Table A–18 Relationship Table

Field Name	Data Type/Size	Description
RELATIONSHIP_NAME	Varying string, 255 bytes	Name describing the relationship of the 2 related events.
CREATOR_FACILITY_ NAME	Varying string, 27 bytes	The facility name that created this relationship.
CREATOR_FACILITY_ VERSION	Varying string, 10 bytes	The version of the facility that created this relationship.
SOURCE_FACILITY_NAME	Varying string, 27 bytes	The source facility name.
SOURCE_EVENT_NAME	Varying string, 15 bytes	Name describing the source event.
SOURCE_ITEM_NAME	Varying string, 15 bytes	Name describing the source item.
TARGET_FACILITY_NAME	Varying string, 27 bytes	The target facility name.
TARGET_EVENT_NAME	Varying string, 15 bytes	Name describing the target event.
TARGET_ITEM_NAME	Varying string, 15 bytes	Name describing the target item.
COMMENT	Varying string, 80 bytes	The comment describing the relationship.
HIERARCHY_LEVEL	Signed longword	Hierarchy level for the relationship. Value of zero indicates no hierarchy level specified.
S1	Varying string, 255 bytes	Character string value for the S1 parameter.

Bold signifies key fields.

A.4 Relations for the Collected Data

A relation for the data collected for an event is created only if data has been collected for that event. The name for an event data relation is: EPC$<SELECTION_RECORD_ID>_<FACILITY_NUMBER>_<EVENT_ NAME>. For example, the Oracle Rdb TRANSACTION event is EPC$1_221_TRANSATION. See the documentation for the facilities with which you are working for more detail.

For a point event data relation, the item name of an item is used as the column name in the relation.

For a duration event data relation, the suffixes _START and _END are added to the item names for items that are collected at event start time and event end time, respectively.

For example, an item named MEMORY_USAGE will have a column name MEMORY_USAGE in a point event data relation. A similar item will have a column name MEMORY_USAGE_START for item data collected at event start time. A similar item will have a column name MEMORY_USAGE_END for item data collected at event end time.

In addition to facility-defined events and process items, all event relations will also include the fields described in Table A–19.

Table A–19 Event Relation Fields

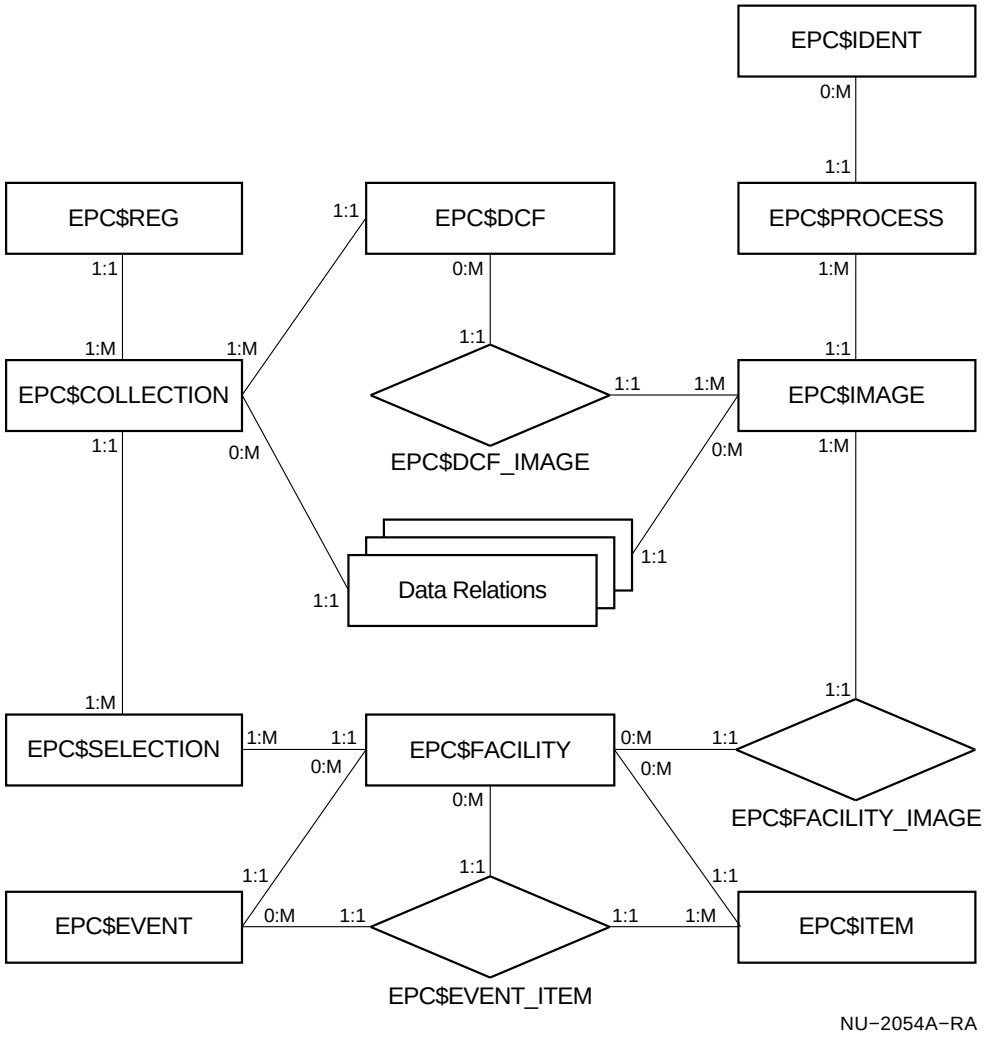
Field Name	Data Type/Size	Description
COLLECTION_RECORD_ID	Signed word	The EPC$COLLECTION record identifier, a foreign key.
IMAGE_RECORD_ID	Signed longword	The EPC$IMAGE record identifier, a foreign key.
CONTEXT_NUMBER	Signed longword	The context number.
TIMESTAMP_POINT	Date	Point event logged time, for point events only.
TIMESTAMP_START	Date	Duration event start time, for duration events only.
TIMESTAMP_END	Date	Duration event end time, for duration events only.
ELAPSED	Date	Elapsed time of a duration event.

Italic signifies foreign key fields.

A.5 The Entity Relationship Diagram for the Database

Figure A–1 shows the ER diagram for the formatted database.

Figure A–1 The Entity Relationship Diagram for the Database



A.6 The Predefined Views

To facilitate data retrieval, the views described in the following sections are predefined:

A.6.1 Relating Collection Information with Event Data

A view is defined for each event relation to relate collection information to the events:

```
CREATE VIEW <Event RELATION_NAME>_C
AS
SELECT *
FROM   <Event RELATION_NAME>, EPC$COLLECTION
WHERE  <Event RELATION_NAME>.COLLECTION_RECORD_ID =
       EPC$COLLECTION.COLLECTION_RECORD_ID
```

A.6.2 Relating Selection and Collection Information with Event Data

A view is defined for each event relation to relate selection and collection information to the events:

```
CREATE VIEW <Event RELATION_NAME>_SC
AS
SELECT *
FROM   <Event RELATION_NAME>, EPC$COLLECTION, EPC$SELECTION
WHERE  <Event RELATION_NAME>.COLLECTION_RECORD_ID =
       EPC$COLLECTION.COLLECTION_RECORD_ID
  AND  EPC$COLLECTION.SELECTION_RECORD_ID = EPC$SELECTION.SELECTION_RECORD_ID
```

A.6.3 Relating Image Information with Event Data

A view is defined for each event relation to relate image information to the events:

```
CREATE VIEW <Event RELATION_NAME>_I
AS
SELECT *
FROM   <Event RELATION_NAME>, EPC$IMAGE
WHERE  <Event RELATION_NAME>.IMAGE_RECORD_ID = EPC$IMAGE.IMAGE_RECORD_ID
```

A.6.4 Relating Process and Image Information with Event Data

A view is defined for each event relation to relate process and image information to the events:

```
CREATE VIEW <Event RELATION_NAME>_PI
AS
SELECT *
FROM   <Event RELATION_NAME>, EPC$IMAGE, EPC$PROCESS
WHERE  <Event RELATION_NAME>.IMAGE_RECORD_ID = EPC$IMAGE.IMAGE_RECORD_ID
  AND  EPC$IMAGE.PROCESS_RECORD_ID = EPC$PROCESS.PROCESS_RECORD_ID
```

A.6.5 Limitations

Due to the range limitation placed on SELECTION_RECORD_ID and the sizes of COLLECTION_RECORD_ID, IDENT_RECORD_ID, PROCESS_RECORD_ID, and IMAGE_RECORD_ID, the maximum numbers of unique record that can be stored in EPC$SELECTION, EPC$COLLECTION, EPCDCF, EPCIDENT, EPC$PROCESS, and EPC$IMAGE relations are 999, 32768, 32768, 32768, 32768, and 2147483647, respectively.

B

VAX RMS File Format

This appendix describes the formatted VAX RMS file created by Oracle Trace when you specify the /TYPE=RMS qualifier to the FORMAT command.

The Oracle Trace formatter component merges, formats, and stores the event data collected from multiple collections of the same facility selection into a formatted VAX RMS file. Formatted VAX RMS files are sequential files with variable length records. Maximum size of a record is 32,765 bytes. This is the maximum record size for a VAX RMS sequential file written to disk.

Only collections with identical facility selections can be merged into a single VAX RMS file. Identically defined selections have the same data for the following fields:

- For the COLLECTION record—SELECTION NAME and SELECTION COMMENT
- For the FACILITY record—FACILITY NUMBER, FACILITY NAME, FACILITY VERSION, FACILITY DEFINITION TIME, and COLLECTION CLASS
- For the EVENT record—FACILITY NUMBER, EVENT NUMBER, EVENT NAME, EVENT HEADER, EVENT TYPE, RESOURCE ITEM FLAGS, and everything in each COLLECTION ITEM DESCRIPTOR

B.1 Data Types

The following sections describe the data types used in the formatted VAX RMS file.

B.1.1 Date

A date field contains a time and date in the standard OpenVMS quadword binary format.

B.1.2 Fixed ASCIC/ASCIW String

A fixed ASCIC string field records the actual length of the string in the first byte, followed by the actual string itself. A fixed ASCIW string field records the actual length of the string in the first word, followed by the actual string itself. The field size is fixed. The content of the additional space following the actual string is undetermined. For example, the SELECTION NAME field of the COLLECTION record is a field of 33 bytes long. The first byte of this field contains a count of the selection name, followed by the selection name of up to 32 characters. Therefore, for a selection name that is only five characters long, the first byte of this field has a value of five, followed by the selection name, followed by 27 unused bytes with unpredictable values in them.

B.2 Records Organization

There are ten types of records: seven for control information (FMTDICT, COLLECTION, FACILITY, DCF, IMAGE, FACILITY-REGISTRATION, and EVENT records) and three for collected data (POINT, START, and END records). The first byte of a record contains a literal which indicates the type of the record. Table B–1 shows the possible record types.

Table B–1 Record Type Literal

Record Type	Literal
FMTDICT	EPC$K_FMTDICT_REC
COLLECTION	EPC$K_COLLECTION_REC
FACILITY	EPC$K_FACILITY_REC
DCF	EPC$K_DCF_REC
IMAGE	EPC$K_IMAGE_REC
FACILITY_REGISTRATION	EPC$K_FACREG_REC
EVENT	EPC$K_EVENT_REC
POINT	EPC$K_POINT_REC
START	EPC$K_START_REC
END	EPC$K_END_REC

The formatter dictionary record (FMTDICT) contains dictionary information that is used by the formatter only. Instances of this record may appear anywhere in a formatted VAX RMS file. This record should simply be ignored and skipped over by any program that reads the VAX RMS file.

A formatted VAX RMS file consists of one or more collection sections. Each collection section has a COLLECTION record, followed by one or more FACILITY records and a collected data section. The appearance of a new COLLECTION record in a file signifies the end of the last collection section and the beginning of the next, and the information in the new COLLECTION record supersedes that of the previous one.

A collected data section is composed of one or more merged data collection files (DCFs). A DCF record signifies the end of the previous DCF and the beginning of a new one. Within a collected data section are a number of IMAGE records, FACILITY-REGISTRATION records, EVENT records, and the data collection records. An IMAGE record that describes an image precedes the data that was collected from that image. A FACILITY-REGISTRATION record indicates the registration of a facility through an image. The EPID in this record points to the last IMAGE record that has the same EPID. An EVENT record that describes an event precedes the first collected data record for that event. The last IMAGE record that has an EPID that matches the EPID field of a collected data record shows the image where the data was collected.

Note that data collected from the same collection may be separated by collection sections of some other collections, depending on the order in which the DCFs were merged. In this case, each of the separated parts is a complete collection section with an identical COLLECTION record, and most likely the same set of FACILITY and IMAGE records. They would have different DCF records, however.

B.2.1 Control Records

The control records are described in the following sections.

B.2.1.1 FMTDICT Record

Table B–2 describes the fields in the formatter dictionary record.

Table B–2 FMTDICT Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_FMTDICT_ REC to indicate that this is a formatter dictionary record.
DICTIONARY SIZE	Signed longword	4	The size of the dictionary to follow.

(continued on next page)

Table B–2 (Cont.) FMTDICT Record

Field Name	Data Type	Bytes	Description
DICTIONARY	N bytes		The formatter dictionary.

B.2.1.2 COLLECTION Record

Table B–3 describes the fields in the COLLECTION record.

Table B–3 COLLECTION Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_ COLLECTION_REC to indicate that this is a COLLECTION record.
SELECTION NAME	Fixed ASCIC string	33	The selection name.
SELECTION COMMENT	Fixed ASCIC string	81	The comment for the selection.
COLLECTION NAME	Fixed ASCIC string	33	The collection name.
START TIME	Date	8	The collection start time and date.
END TIME	Date	8	The collection end time and date.
REG ID LIST COUNT	Signed longword	4	Registration ID list count.
REG ID	Fixed ASCIC string	256	The registration ID.

B.2.1.3 FACILITY Record

Table B–4 describes the fields in the FACILITY record.

Table B–4 FACILITY Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_ FACILITY_REC to indicate that this is a FACILITY record.

(continued on next page)

Table B–4 (Cont.) FACILITY Record

Field Name	Data Type	Bytes	Description
FACILITY NUMBER	Signed word	2	The facility number.
FACILITY NAME	Fixed ASCIC string	28	The facility name.
FACILITY VERSION	Fixed ASCIC string	11	The facility version.
FACILITY DEFINITION TIME	Date	8	Facility definition creation time.
COLLECTION CLASS	Fixed ASCIC string	33	The collection class name.

B.2.1.4 DCF Record

Table B–5 describes the fields in the data collection file record.

Table B–5 DCF Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_DCF_REC to indicate that this is a data collection file information record.
ACTIVATION TIME	Date	8	Collection activation time.
SYSTEM ID	Integer	6	The system identification of the node, obtained with a SYS$GETSYI call, specifying SYI$_NODE_SYSTEMID.
TOTAL FILES	Signed word	2	The total number of DCFs from the collection.
FILE NUMBER	Signed word	2	The file number of this DCF.
FILE NAME	Fixed ASCIC string	256	The file name of this DCF.

B.2.1.5 IMAGE Record

Table B–6 describes the fields in the IMAGE record.

Table B–6 IMAGE Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_IMAGE_ REC to indicate that this is an IMAGE record.
PRODUCT VERSION	Fixed ASCIC string	11	The Oracle Trace product version string.
DCF VERSION	Fixed ASCIC string	11	The data collection file version string.
FORMAT VERSION	Fixed ASCIC string	11	The formatted VAX RMS file version string.
EPID	Signed longword	4	The extended process identity number.
UIC	Signed longword	4	The process user identification.
ACCOUNT	Fixed ASCIC string	9	The process account name.
USERNAME	Fixed ASCIC string	13	The user name of the process.
PROCESS NAME	Fixed ASCIC string	16	The process name.
PROCESS CREATION TIME	Date	8	Process creation time.
BASE PRIORITY	Signed longword	4	The base priority of the process.
PROCESS MODE	Signed longword	4	The process mode.
SYSTEM ID	Integer	6	The system identification of the node, obtained with a SYS$GETSYI call, specifying SYI$_NODE_SYSTEMID.
NODE NAME	Fixed ASCIW string	406	The DECnet node name, obtained with a SYS$GETSYI call, specifying SYI$_ NODENAME.
CPU TYPE	Fixed ASCIC string	32	The CPU type, obtained with a SYS$GETSYI call, specifying SYI$_HW_NAME.

(continued on next page)

Table B–6 (Cont.) IMAGE Record

Field Name	Data Type	Bytes	Description
VMS VERSION	Fixed ASCIC string	9	The OpenVMS version, obtained with a SYS$GETSYI call, specifying SYI$_ VERSION.
SYSTEM BOOT TIME	Date	8	The time the node was booted, result of an SYS$GETSYI call when specifying SYI$_ BOOTTIME.
IMAGE LINK TIME	Date	8	The time when this image was linked.
IMAGE ID	Fixed ASCIC string	16	The image identity number.
IMAGE NAME	Fixed ASCIC string	256	The full file specification of the image.
IMAGE ACTIVATION TIME	Date	8	Image activation time.

B.2.1.6 FACILITY-REGISTRATION Record

Table B–7 describes the fields in the FACILITY-REGISTRATION record.

Table B–7 FACILITY-REGISTRATION Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_FACREG_ REC to indicate that this is an IMAGE record.
EPID	Signed longword	4	The extended process identity number.
FACILITY NUMBER	Signed word	2	The facility number.
FACILITY VERSION	Fixed ASCIC string	11	The facility version.
REG ID	Fixed ASCIC string	256	The registration ID.

B.2.1.7 EVENT Record

Table B–8 describes the fields in the EVENT record.

Table B–8 Event Record Description

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains EPC$K_EVENT_REC to indicate that this is an EVENT record.
FACILITY NUMBER	Signed word	2	The facility each event is associated with.
EVENT NUMBER	Signed byte	1	The event number for this event.
EVENT NAME	Fixed ASCIC string	16	Name describing the event.
EVENT HEADER	Fixed ASCIC string	16	The column header used in reports.
EVENT TYPE	Signed byte	1	The event type. Possible values are: • EPC$K_EVENT_TYPE_POINT(81)—Indicates a point event type. • EPC$K_EVENT_TYPE_START(82)—Indicates a duration event type, with start items in the ITEM DESCRIPTOR LIST. • EPC$K_EVENT_TYPE_END(83)—Indicates a duration event type, with end items in the ITEM DESCRIPTOR LIST.
ITEM DESCRIPTOR LIST COUNT	Signed longword	4	Item descriptor list count.
ITEM DESCRIPTOR LIST	44n bytes		A list of item descriptor structures. See Section B.2.1.8.

B.2.1.8 Item Descriptor

Table B–9 describes the fields in the item descriptor record.

Table B–9 Item Descriptor

Field Name	Data Type	Bytes	Description
ITEM NUMBER	Signed byte	1	The item number for this item.
ITEM NAME	Fixed ASCIC string	16	Name describing the item.
ITEM HEADER	Fixed ASCIC string	16	The column header used in reports.
ITEM TYPE	Signed word	2	The data type of the item.
ITEM WIDTH	Signed word	2	The column width used in reports.
ITEM MAX SIZE	Signed word	2	The maximum size of the item.
ITEM USAGE	Signed byte	1	Usage of the item. Possible values follow: • EPC$K_ITM_U_LEVEL(1)—Indicates the item is a level. A level is a meter or gauge that indicates the "CURRENT" value of some metric. Its value may go up and down. • EPC$K_ITM_U_COUNTER(2)—Indicates the item is a counter. A counter indicates the number of times something occurred. Its value is ever increasing. • EPC$K_ITM_U_PERCENT(3)—Indicates the item is a percentage. A percentage is a type of level. • EPC$K_ITM_U_TEXT(4)—Indicates the item is text. • EPC$K_ITM_U_PRIVATE(5)—Indicates the item is for private use by the facility.

(continued on next page)

Table B–9 (Cont.) Item Descriptor

Field Name	Data Type	Bytes	Description
ITEM CHARACTERISTICS	Signed longword	4	Item value characteristics. Possible values follow: • EPC$K_ITM_CHR_PRINT(1)—Printable item value. • EPC$K_ITM_CHR_NONPRINT(2)—Non-printable item value.

B.3 Data Collection Records

The data collection records contain a fixed length portion with fields that point to the image and facility that generated the data. This is followed by the actual collected data. The optional process data items appears first, followed by the facility data items in the order the collection item descriptors appear in each EVENT record. Each data item occupies only as much space as needed according to the data type: four bytes for a longword, nine bytes for a ASCIC type with a count of eight in the count field and so forth.

Table B–10 describes the fields in the data collection record.

Table B–10 Data Collection Record

Field Name	Data Type	Bytes	Description
RECORD TYPE	Signed byte	1	Contains one of the following to indicate one of the data collection records: • EPC$K_POINT_REC—Indicates a point data collection record. • EPC$K_START_REC—Indicates a start data collection record. • EPC$K_END_REC—Indicates an end data collection record.

(continued on next page)

Table B–10 (Cont.) Data Collection Record

Field Name	Data Type	Bytes	Description
EPID	Signed longword	4	The extended process identity number.
TIME	Date	8	The record log time and date.
FACILITY NUMBER	Signed word	2	The facility that logged the event.
EVENT NUMBER	Signed byte	1	The event number of the logged event.
CONTEXT NUMBER	Signed longword	4	Context number.
HANDLE	Signed longword	4	Instance handle used to match start and end records.
COLLECTED DATA	Varied structure	n	The collected data.

Index

@ (Execute Procedure) command, 7–3

C

CDD/Repository, 2–1
Commands, 7–1tab
  @ (Execute Procedure), 7–3
  FORMAT, 7–4
  HELP, 7–11
  REPORT, 7–12

D

Data merging, 2–1
Detail report, 3–21
Duration events
  in Summary reports, 3–7

F

FORMAT command
  examples
    Oracle Rdb database, 2–2
    VAX RMS file, 2–4
  format, 7–4
Frequency report, 3–24

H

HELP command, 7–11

M

Merging data files, 2–1

R

REPORT command
  format, 7–12
  options
    EVENT, 7–20
    ITEM, 7–24
    JOIN, 7–25
    RESTRICTION, 7–26

S

Summary report, 3–4
  duration events in, 3–7