Oracle® Trace for OpenVMS

Release Notes

Release 7.4.1.0

August 2024

Oracle Trace Release Notes, Release 7.4.1.0 for OpenVMS Industry Standard 64 for Integrity Servers and OpenVMS Alpha

Oracle Corporation - Worldwide Headquarters, 2300 Oracle Way, Austin, TX 78741, United States

Primary Author: Rdb Engineering and Documentation group

Contents

6 Documentation Corrections, Additions and Changes

7 Known Problems and Restrictions

Tables

Preface

Purpose of This Manual

This manual contains release notes for Oracle Trace release 7.4.1.0. The notes describe changed and enhanced features; upgrade and compatibility information; new and existing software problems and restrictions; and software and documentation corrections.

Intended Audience

This manual is intended for use by all Oracle Trace users. Read this manual before you install, upgrade, or use Oracle Trace release 7.4.1.0.

Access to Oracle Support

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

Reader 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.

- 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?

If you find any errors or have any other suggestions for improvement, please indicate the document title, release date, chapter, section, and page number (if available).

Please direct all comments, and corrections to this email address: **infordb_us@oracle.com**.

If you have problems with the software, please contact your local Oracle Support Services.

Online Document Format

You can view the Adobe Acrobat formatted documentation using the Acrobat Reader, which allows anyone to view, navigate, and print documents in the Adobe Portable Document Format (PDF). See http://www.adobe.com for information about obtaining a free copy of Acrobat Reader and for information on supported platforms.

The Oracle Rdb Server documentation in Adobe Acrobat format can be reached from the Oracle Rdb Server main web page. At the bottom of the page are links to documentation for all of the Rdb releases. Following is the link for the Oracle Rdb Server main page:

```
https://www.oracle.com/database/technologies/related/rdb.html
```

Optional Software

The following optional products may be referenced in these release notes:

- Oracle CDD/Repository
- Oracle Replication Option
- Oracle SQL/Services & OCI Services for Oracle Rdb
- Oracle SQL/Services Client API Kits
- Oracle ODBC Driver for Rdb
- Oracle JDBC for Rdb
- Oracle Rdb Developer Tools for Visual Studio
- Oracle Rdb Extension for SQL Developer
- Oracle Rdb Connectivity Manager

Document Structure

This manual consists of the following chapters:

Chapter 1	Describes how to install Oracle Trace release 7.4.1.0.
Chapter 2	Describes problems corrected in Oracle Trace release 7.4.1.0.
Chapter 3	Describes problems corrected in Oracle Trace release 7.2.0.3.
Chapter 4	Describes problems corrected in Oracle Trace release 7.2.0.2.
Chapter 5	Describes problems corrected in Oracle Trace release 7.2.0.1.
Chapter 6	Provides information not currently available in the Oracle Trace documentation set.
Chapter 7	Describes problems, restrictions, and workarounds known to exist in Oracle Trace release 7.4.1.0

1
Installing Oracle Trace Release 7.4.1.0

1.1 Installation Instructions

Prior to the installation, you must execute the following command on each node in the cluster that shares the same EPC$ADMIN_DB:

```
$ COLLECT STOP SYSTEM/ABORT
```

This software update is installed using the OpenVMS VMSINSTAL utility.

Install the kit logging in to suitably privileged account and entering the following command for an Integrity system:

```
$ @SYS$UPDATE:VMSINSTAL EPC07410I074 [saveset-location]
```

or entering the following command for an Alpha system:

```
$ @SYS$UPDATE:VMSINSTAL EPC07410A074 [saveset-location]
```

The saveset-location is a disk directory that contains the kit saveset.

After the installation, execute the command `@SYS$STARTUP:EPC$STARTUP.COM` to restart the EPC$REGISTRAR process.

Refer to the *Oracle Trace for OpenVMS Installation Guide* for release 7.4 for complete installation instructions.

1.2 Requirements

The following conditions must be met in order to install this software:

- Oracle Rdb Server release 7.2 or later version must be used for Oracle Trace databases. It is only the HISTORY and ADMIN databases stored in EPC$DATABASE_DIR which are managed using such versions.

  Note that applications may use other versions in a multi-version Rdb environment, such as Oracle Rdb Server release 7.4.
- This Oracle Trace release requires the following OpenVMS environments:
  - OpenVMS Alpha version 8.4 or later.
  - OpenVMS Industry Standard 64 version 8.4 or later.
- Some hardware configurations require the installation of OpenVMS V8.4-1H1 or later versions from VMS Software Inc. (VSI).
- Oracle strongly recommends that all available OpenVMS patches are installed on all systems prior to installing Oracle Trace. Contact your HPE or VSI support representative for more information and assistance.

2
Software Errors Fixed in Oracle Trace Release 7.4.1.0

This chapter describes software errors that are fixed by Oracle Trace release 7.4.1.0.

2.1 Software Errors Fixed in Oracle Trace Release 7.4.1.0

2.1.1 EPC$REGISTRAR Startup Access Violation if Unable to Access Rdb RUJ file

There was a problem where the Oracle Trace EPC$REGISTRAR process failed with a SYSTEM-F-ACCVIO and an EPC$BUGCHECK.DMP file was created because an Oracle Rdb Server RUJ (recovery unit journal) file could not be accessed for transactions involving the Oracle Trace EPC$ADMIN_DB or EPC$HISTORY_DB databases.

This could happen due to disk quotas being exceeded or if there was an invalid RUJ file directory specification. For example, logical names such as RDM$RUJ, SYS$SCRATCH, or logical names and devices that those logical names reference are no longer valid or available.

The Oracle Trace EPC$HOME_DIR:EPC$REGISTRAR_nodename.LOG showed the access violation and indicated that an EPC$BUGCHECK.DMP file was created but did not output an error message indicating the cause of the problem. The EPC$BUGCHECK.DMP file also did not indicate the actual cause of the problem.

This problem has been fixed. Now, instead of a general SYSTEM-F-ACCVIO error and an unnecessary EPC$BUGCHECK.DMP file being created a specific error message will be output which indicates the cause of the problem.

The following example shows the problem happening when the Oracle Trace Registrar process was started and an RUJ file could not be accessed for a transaction involving the Oracle Trace EPC$ADMIN_DB database.

```
$ RUN SYS$SYSTEM:EPC$REGISTRAR.EXE
```

The EPC$HOME_DIR:EPC$REGISTRAR_nodename.LOG showed:

```
%EPC-E-BUGCHK, Fatal error encountered, a dump file is being generated
%SYSTEM-F-ACCVIO, access violation, reason mask=01, virtual
address=0000000000000000, PC=0000000000000000, PS=0000001B
```

```
 Improperly handled condition, image exit forced by last chance handler.
   Signal arguments:   Number = 0000000000000005
                       Name   = 000000000000000C
                                0000000000000001
                                0000000000000000
                                0000000000000000
                                000000000000001B

   Register dump:
   .
   .
   .
```

The EPC$BUGCHECK.DMP file showed:

```
Stack Dump Summary
***** Exception at 00000000001B3480 : EPC$REGISTRAR\SQL$SIGNAL_STOP
   Module SQL$GETERR + 00001D60; line 6780
```

The following example shows the expected diagnostics. The error here indicates that the default device for the account is undefined, or possibly defined as a concealed logical that is not defined consistently across the cluster.

```
$ RUN SYS$SYSTEM:EPC$REGISTRAR.EXE
%RDB-F-SYS_REQUEST, error from system services request
-RDMS-F-FILACCERR, error parsing file
    DISK_DEV:[RDM$RUJ]EPC$ADMIN_DB$04C7MNU3G05U2H3Q0G07M5QCNT.RUJ;
-COSI-I-NOTDISKFILE, file is not a disk file
```

This problem has been corrected in Oracle Trace release 7.4.1.0.

2.1.2 Unexpected Looping When Using the BLR Converter

Bug 30401959

It was possible that SYS$SYSTEM:EPC_BLR_CONVERTER.EXE (usually run via the procedure EPC_BLR_TO_SQL_CONVERTER.COM from EPC$EXAMPLES) could enter a loop when processing some queries. This most likely occurred when a query contained references to COUNT(value_expression) or COUNT(DISTINCT value_expression) aggregate functions. The internal format of these functions changed in Oracle Rdb Server release 7.3 and later versions. See Section 7.1.2 for more details.

This problem has been corrected in Oracle Trace release 7.4.1.0. The query (BLR) dumper has been corrected to avoid this problem in the future.

3
Software Errors Fixed in Oracle Trace Release 7.2.0.3

This chapter describes software errors that are fixed by Oracle Trace Release 7.2.0.3.

3.1 Software Errors Fixed in Oracle Trace Release 7.2.0.3

3.1.1 Removal of Four Million Block Size Limit for Oracle Trace Data Collection Files

The size limit of four million blocks for individual Oracle Trace Data Collection files has been eliminated. The limitation was based on certain 32 bit values used for accessing the Data Collection files now being carried in 64 bits. This change did not require any changes to existing Oracle Trace commands or functionality. Single or multiple Data Collection files can still be used for Oracle Trace data collections and a size limitation can still be specified for Data Collection files.

The new supported limit in the size of a single Oracle Trace Data Collection file is now 2,147,483,647 blocks, which is also the maximum block size limit that can be specified by the /COLLECTION_FILES=(MAX_ALLOCATION=blocks) command qualifier on the COLLECT SCHEDULE COLLECTION command. The actual attainable maximum block size is limited by available disk space and any OpenVMS resource restrictions such as maximum file size or device limitations.

Previously the only way to avoid the four million block limit on individual Oracle Trace Data Collection files was to specify that multiple Data Collection files be used for an Oracle Trace collection.

This problem has been corrected in Oracle Trace release 7.2.0.3.

3.1.2 Missing Support for Recent SQL Syntax in BLR Converter

Prior releases of the BLR converter did not support the following SQL language features:

- statistics functions VARIANCE (this also includes VAR_POP, and VAR_SAMP) and STDDEV (this also includes STDDEV_POP, and STDDEV_SAMP)
- built-in functions CONCAT, BITSTRING, GREATEST, LEAST, ROUND, TRANSLATE and TRUNC
- EXCEPT, MINUS, INTERSECT operators
- special functions SYSTEM_UID, SESSION_UID, CURRENT_UID and UID
- support for sequence pseudo functions CURRVAL, and NEXTVAL

- OFFSET clause of the SELECT statement
- LOCK TABLE, COMMIT AND CHAIN and ROLLBACK AND CHAIN statement
- DECLARE variables in stored procedures
- multiple JOIN clauses are now formatted correctly
- support for UNION ALL and derived tables with more than 255 select expressions

3.1.3 Unexpected Looping When Using the BLR Converter

Bug 8881681

It was possible that SYS$SYSTEM:EPC_BLR_CONVERTER.EXE (usually run via the procedure EPC_BLR_TO_SQL_CONVERTER.COM from EPC$EXAMPLES) could enter a loop when processing some queries. This mostly likely occurred when the query contain call to such functions as CONCAT (including the || operator), GREATEST, LEAST, ROUND or TRUNC. The problem occurred during the problem reporting to the associated .LOG file.

This problem has been corrected in Oracle Trace release 7.2.0.3. The query (BLR) dumper has been corrected to avoid this problem in future.

3.1.4 Various Memory Leaks Corrected

When using various report generation and analysis functions, a process could unexpectedly consume all of P0 virtual address space and fail with memory allocation errors.

This problem has been corrected in Oracle Trace release 7.2.0.3. Several memory leaks have been corrected. Memory usage has been generally reduced and performance improved in many cases.

3.1.5 Possible Data Collection File Corruption

In rare cases, an Oracle Trace data collection file could be incorrectly written resulting in possible EPC-E-FMT_DCF_FAILURE message while formatting the data collection file.

This problem has been corrected in Oracle Trace release 7.2.0.3.

3.1.6 File Size Limit on Collection did not Properly End Collection

In prior releases of Oracle Trace, when using the /COLLECTION_FILES=MAX_ALLOCATION=n qualifier, Oracle Trace did not correctly terminate the collection when the collection file size reached the maximum (though the file was no longer being written to).

This problem has been corrected in Oracle Trace release 7.2.0.3.

4
Software Errors Fixed in Oracle Trace Release 7.2.0.2

This chapter describes software errors that are fixed by Oracle Trace release 7.2.0.2.

4.1 Software Errors Fixed in Oracle Trace Release 7.2.0.2

4.1.1 Unexpected SQL-F-NOTXNOUT Error If /ELAPSED_TIME Used for FORMAT command

There was a problem in Oracle Trace where an %SQL-F-NOTXNOUT error was output and a dump file was generated when the ELAPSED_TIME qualifier was specified with the FORMAT command.

The ELAPSED_TIME qualifier can be used to calculate the elapsed time for each duration event and have the result stored in the Oracle Rdb Server database which the FORMAT command creates.

The problem was that at the very end of the format operation, after the formatting of the Oracle Trace Data Collection file(s) to the Rdb database was done, the code which calculates and stores the elapsed time for duration events to the database tried to commit a database transaction when none was previously started. This caused the format operation to abort with the %SQL-F-NO_TXNOUT error and the dump. This problem has been fixed,

The following example shows the problem. The problem occurs near the completion of formatting the Oracle Trace Data Collection file COLLECTION.DAT to the Rdb database FORMATDB.RDB.

```
$ COLLECT FORMAT /ELAPSED/TYPE=RDBVMS COLLECTION.DAT FORMATDB
%EPC-I-FMT_RDB_CREATE, Creating database FORMATDB
%EPC-S-FMT_RDB_SUCCESS, Successfully created database
%EPC-I-FMT_DCF_BEGIN, Formatting data file COLLECTION.DAT
%EPC-S-FMT_DCF_SUCCESS, Successfully formatted data file COLLECTION.DAT
%EPC-E-BUGCHK, Fatal error encountered, a dump file is being generated
%SQL-F-NO_TXNOUT, No transaction outstanding
%EPC-E-OPFAIL, Operation failed
```

The following example shows that the problem has been fixed and the Oracle Trace Data Collection file COLLECTION.DAT is now succesfully formatted to the Rdb database FORMATDB.RDB.

```
$ COLLECT FORMAT /ELAPSED/TYPE=RDBVMS COLLECTION.DAT FROMATDB
%EPC-I-FMT_RDB_CREATE, Creating database FROMATDB
%EPC-S-FMT_RDB_SUCCESS, Successfully created database
%EPC-I-FMT_DCF_BEGIN, Formatting data file COLLECTION.DAT
%EPC-S-FMT_DCF_SUCCESS, Successfully formatted data file COLLECTION.DAT
%EPC-S-FMT_SUCCESS, Formatting successfully completed
```

The only way to avoid this problem is not to specify the ELAPSED_TIME qualifier with the FORMAT command.

This problem has been corrected in Oracle Trace release 7.2.0.2.

5
Software Errors Fixed in Oracle Trace Release 7.2.0.1

This chapter describes software errors that are fixed by Oracle Trace Release 7.2.0.1.

5.1 Software Errors Fixed in Oracle Trace Release 7.2.0.1

5.1.1 Invalid Oracle Trace Summary Report Elapsed Time Statistics

There was a problem with the *Summary Report Elapsed Time* statistics output by Oracle Trace on OpenVMS IA-64 which caused incorrect *Elapsed Time* values to be displayed. These values could consist of large positive or large negative numbers or the value "NaNQ".

This happened because Oracle Trace on the OpenVMS Industry Standard 64 for Integrity Servers uses IEEE floating point but a system service was being used to calculate the *Elapsed Time* statistics which did not expect IEEE floating point values. This problem has been corrected and the correct *Elapsed Time* statistical values will now be output for Oracle Trace Summary reports. Note that this problem only happened for *Summary Report Elapsed Time* statistics.

The following example of this problem shows that the column for the Elapsed Time statistics in an Oracle Trace Summary report had incorrect values that were too large or negative.

```
$ set verify
$ @reportrdb.com
COLLECT REPORT RDB_DB-
    /TYPE=SUMMARY-
    /TITLE="RDB/VMS SUMMARY REPORT"-
    /STATISTICS=ALL-
    /OUTPUT=RDB_SUMMARY.TXT
%EPC-S-RPCL_SUCCESS, Report successfully completed
$ type RDB_SUMMARY.TXT

9-MAY-2007 17:37             Rdb/VMS Summary Report                 Page 2
Selection: RDB_SELECTION                             Oracle  Trace V7.2-00

Event:  Request Actual  In Facility:  RDBVMS          Version:  V7.2

           Elapsed     AIJ File     As Batch      As Read     As Write
                         Writes        Write        Stall        Stall
Minimum  -40581258            0            0            0            0
Maximum  122303416            0            0            0            0
Mean     -21358556         0.00         0.00         0.00         0.00
Std Dev  918330682         0.00         0.00         0.00         0.00
95 Prct  158634256         0.00         0.00         0.00         0.00
Total    -81162516            0            0            0            0
   .
   .
   .
```

This problem has been corrected in Oracle Trace release 7.2.0.1.

5.1.2 Standard Deviation, 95 Percentile Precision Problems

There was a problem with the *Summary Report Standard Deviation and 95 Percentile* statistics output by Oracle Trace on OpenVMS IA-64 which could cause invalid results if these statistics were being calculated based on large numerical values. Invalid results such as the value "NaNQ" or "NaNQ00" could be output.

This problem happened because the floating point precision used for these calculations was not adequate for large numerical values. This caused too much loss of precision due to rounding of values during the repeated calculations used to determine these statistics. This problem has been fixed and now the proper precision will be used to calculate these statistics.

The following example of this problem shows that the column for the "REQ OPER" statistics in an Oracle Trace Summary report had incorrect values for the Standard Deviation and 95 Percentile statistics. Instead of the correct numerical values, "NaNQ00" was output because the calculation had invalid results since the floating point precision used was too small leading to too much rounding of numerical values during the repeated calculations necessary to determine these statistics.

```
$ set verify
$ @reportrdb.com
COLLECT REPORT RDB_DB-
    /TYPE=SUMMARY-
    /TITLE="RDB/VMS SUMMARY REPORT"-
    /STATISTICS=(STANDARD_DEVIATION,95_PERCENTILE)-
    /OUTPUT=RDB_SUMMARY.TXT
%EPC-S-RPCL_SUCCESS, Report successfully completed
$ type RDB_SUMMARY.TXT
9-MAY-2007 17:37             Rdb/VMS Summary Report               Page 1
Selection: RDB_SELECTION                            Oracle Trace V7.2-00

Event:  Database        In Facility:  RDBVMS          Version:  V7.2

          Lock Stall   Pagefaults    Pagefault         Prom     Req Oper
                Time                      I/Os    Deadlocks

Std Dev      2263.05        17.25         0.55         0.00      NaNQ00
95 Prct      4961.93        42.81         1.34         0.00      NaNQ00
   .
   .
   .
```

The only way to avoid this problem is to limit the collection time used to gather the statistics to a shorter duration so smaller numerical values will be used to determine these statistics.

This problem has been corrected in Oracle Trace release 7.2.0.1.

6

Documentation Corrections, Additions and Changes

This chapter describes problems, restrictions and additional documentation relating to Oracle Trace version 7.4.1.0.

6.1 Clarification on SQL_STRING Usage When Using EPC_BLR_TO_SQL_CONVERTER.COM

Bug 18762712

A customer reported that the saved SQL strings were truncated to 1024 characters. It wasn't clearly documented that this column size could be changed by altering the EPC$EXAMPLES:EPC_BLR_TO_SQL_CONVERTER.COM procedure or running the image (SYS$SYSTEM:EPC_BLR_CONVERTER.EXE) directly.

Oracle Trace provides an image to process the BLR (query definition) stored by the collector. The procedure EPC_BLR_TO_SQL_CONVERTER.COM in EPC$EXAMPLES can be used to run this tool.

The EPC_BLR_CONVERTER.EXE image accepts a number of options and arguments to control the conversion.

EPC_BLR_CONVERTER.EXE Options

The options needed for the BLR parser are documented below and a typical parse command is shown.

Option	Action	Description
-d	Marks all images as dynamic.	When doing an extract, this directs the scanner to treat all images as dynamic.
-e	formatted database name	In this command procedure, the EPC$FORMAT_DB logical is defined to the formatted database.
-l	log progress	You will be updated as to the current request being processed and how many are left. This shows you the command is indeed working. This is optional.

Option	Action	Description
-p	edit string size for SQL_STRING column	By specifying a value, you can determine the size of the edit string created for viewing the SQL_STRING. See also the -v option.
-t	transaction type	Values are READ_WRITE or BATCH_UPDATE.
-u	exclude this phase in the processing	For BLR parsing, the 4GL processing step is not needed. For best performance, always tell the parser to skip 4GL.
-v	size of SQL_STRING column	The default size is 1024; if the resulting SQL strings are greater than 1024, they will be truncated. If you know the expected size, you can set the size of the field here. Just for completeness sake we will specify 1024 on our command line as well. It is safe to rerun the parser on the same formatted database, so if your SQL_STRING is too small just increase the size and rerun.
-w	output the SQL strings to the EPC$SQL_QUERIES table.	The value TABLE is required.

The following commands illustrate using the converter as an OpenVMS foreign command.

```
$ define/user rdm$bind_buffers 4000
$ epc_blr = "$SYS$SYSTEM:EPC_BLR_CONVERTER.EXE"
$ epc_blr -e EPC$FORMAT_DB -l -t READ_WRITE -u 4GL -v 4000 -w TABLE -d -p T(60)
$
```

The EDIT STRING specification T(n), used in this example, formats text by wrapping near the nth character on to a new line. Refer to the *Oracle Rdb SQL Reference Manual* for more details on the supported EDIT STRING specification.

The BLR parser requires the file EPC$EXAMPLES:RDBX_GLOBALS.CUSTOM and EPC$EXAMPLES:RDBX_VMS_GLOBALS.INI.

6.2 Revised and Reformatted Version of EPC$MSG.DOC in SYS$HELP

Bug 12585141

The file SYS$HELP:EPC$MSG.DOC is a text file that describes all of the Oracle Trace (EPC facility) messages.

In previous versions of Oracle Trace the installation erroneously provided an out-of-date copy of EPC$MSG.DOC. This release provides a corrected, reformatted, and up-to-date file.

6.3 Find the Transaction Type of a Particular Transaction Within the Trace Database

The table EPC$1_221_TRANSACTION in the formatted Oracle Trace database has a column LOCK_MODE_START of INTEGER (longword) datatype. The values of this column indicate the type of transaction.

Value	Transaction Type
8	Read only
9	Read write
14	Batch update

6.4 Using Oracle Trace Collected Data

Oracle Rdb Server provides Oracle Trace with many details about an executed query. In particular the query name (known as the REQUEST NAME) is passed to Oracle Trace if defined.

The following example illustrates the use of the OPTIMIZE AS clause, and how each query name is reflected in the Oracle Trace database. When a collection has been started the following SQL statements will record the query name for each request.

```
SQL> attach 'file PERSONNEL';
SQL> select last_name, first_name
cont> from employees
cont> optimize as request_one;
   .
   .
   .
SQL> select employee_id
cont> from employees
cont> optimize as request_two;
   .
   .
   .
SQL> select employee_id, city, state
cont> from employees
cont> optimize as request_three;
   .
   .
   .
SQL> select last_name, first_name, employee_id, city, state
cont> from employees
cont> optimize as request_four;
   .
   .
   .
```

Once an Oracle Trace database has been populated from the collection, a query such as the following can be used to display the request names and request types. The request type values are described in Table 6–1, Request Types. The unnamed queries in this example correspond to the queries executed by interactive SQL to validate the names of the tables and columns referenced in the user supplied queries.

```
SQL> select REQUEST_NAME, REQUEST_TYPE, TIMESTAMP_POINT
cont> from EPC$1_221_REQUEST_BLR;
REQUEST_NAME                      REQUEST_TYPE   TIMESTAMP_POINT
                                             1   1-MAY-2024 13:23:27.18
                                             1   1-MAY-2024 13:23:27.77
REQUEST_ONE                                  1   1-MAY-2024 13:23:28.21
REQUEST_TWO                                  1   1-MAY-2024 13:23:56.55
REQUEST_THREE                                1   1-MAY-2024 13:24:57.27
REQUEST_FOUR                                 1   1-MAY-2024 13:25:25.44
6 rows selected
$
```

The next example shows the internal query format (BLR) converted to SQL strings after EPC$EXAMPLES:EPC_BLR_TO_SQL_CONVERTER.COM has been run.

```
SQL> SELECT A.REQUEST_NAME, B.SQL_STRING FROM
cont> EPC$1_221_REQUEST_BLR A,
cont> EPC$SQL_QUERIES B
cont> WHERE A.CLIENT_PC = 0 AND A.SQL_ID = B.SQL_ID;
A.REQUEST_NAME
  B.SQL_STRING
REQUEST_ONE
      SELECT C1.LAST_NAME, C1.FIRST_NAME.        FROM EMPLOYEES C1
 . . .
REQUEST_TWO
      SELECT C1.EMPLOYEE_ID.        FROM EMPLOYEES C1
 . . .
REQUEST_THREE
SELECT C1.EMPLOYEE_ID, C1.CITY, C1.STATE.        FROM EMPLOYEES C1
   .
   .
   .
4 rows selected
$
```

Table 6–1 Request Types

Symbolic Name	Value	Comment
RDB_K_REQTYPE_OTHER	0	A query executed internally by Oracle Rdb Server
RDB_K_REQTYPE_USER_ REQUEST	1	A non-stored SQL statement, which includes compound statements
RDB_K_REQTYPE_PROCEDURE	2	A stored procedure
RDB_K_REQTYPE_FUNCTION	3	A stored function
RDB_K_REQTYPE_TRIGGER	4	A trigger action
RDB_K_REQTYPE_ CONSTRAINT	5	A table or column constraint

6.5 Missing Descriptions for the RDBEXPERT Collection Class

Appendix B in the *Oracle Rdb7 Guide to Database Performance and Tuning* describes the event-based data tables in the formatted database for the Oracle Rdb Server PERFORMANCE and RDBEXPERT collection classes. This section provides the missing tables describing the RDBEXPERT collection class.

Table 6–2 shows the TRANS_TPB table.

Table 6–2 Columns for Table EPC$1_221_TRANS_TPB

Column Name	Data Type	Domain
COLLECTION_RECORD_ID	SMALLINT	COLLECTION_RECORD_ID_DOMAIN
IMAGE_RECORD_ID	INTEGER	IMAGE_RECORD_ID_DOMAIN
CONTEXT_NUMBER	INTEGER	CONTEXT_NUMBER_DOMAIN
TIMESTAMP_POINT	DATE VMS	
CLIENT_PC	INTEGER	
STREAM_ID	INTEGER	
TRANS_ID	VARCHAR(16)	
TRANS_ID_STR_ID	INTEGER	STR_ID_DOMAIN
TPB	VARCHAR(127)	
TPB_STR_ID	INTEGER	STR_ID_DOMAIN

Table 6–3 shows the TRANS_TPB_ST table. An index is provided for this table. It is defined with column STR_ID, duplicates are allowed, and the type is sorted.

Table 6–3 Columns for Table EPC$1_221_TRANS_TPB_ST

Column Name	Data Type	Domain
STR_ID	INTEGER	STR_ID_DOMAIN
SEGMENT_NUMBER	SMALLINT	SEGMENT_NUMBER_DOMAIN
STR_SEGMENT	VARCHAR(128)	

6.6 Missing Descriptions for Tables in the Formatted Database

Some of the columns were missing from the tables in Appendix B in the *Oracle Rdb7 Guide to Database Performance and Tuning*. The complete table definitions are described in this section.

Table 6–4 shows the DATABASE table.

Table 6–4 Columns for Table EPC$1_221_DATABASE

Column Name	Data Type	Domain
COLLECTION_RECORD_ID	SMALLINT	COLLECTION_RECORD_ID_DOMAIN
IMAGE_RECORD_ID	INTEGER	IMAGE_RECORD_ID_DOMAIN
CONTEXT_NUMBER	INTEGER	CONTEXT_NUMBER_DOMAIN
TIMESTAMP_POINT	DATE VMS	
CLIENT_PC	INTEGER	
STREAM_ID	INTEGER	
DB_NAME	VARCHAR(255)	
DB_NAME_STR_ID	INTEGER	STR_ID_DOMAIN
IMAGE_FILE_NAME	VARCHAR(255)	
IMAGE_FILE_NAME_STR_ID	INTEGER	STR_ID_DOMAIN

Table 6–5 shows the REQUEST_ACTUAL table.

Table 6–5 Columns for Table EPC$1_221_REQUEST_ACTUAL

Column Name	Data Type	Domain
COLLECTION_RECORD_ID	SMALLINT	COLLECTION_RECORD_ID_DOMAIN
IMAGE_RECORD_ID	INTEGER	IMAGE_RECORD_ID_DOMAIN
CONTEXT_NUMBER	INTEGER	CONTEXT_NUMBER_DOMAIN
TIMESTAMP_START	DATE VMS	
TIMESTAMP_END	DATE VMS	
DBS_READS_START	INTEGER	
DBS_WRITES_START	INTEGER	
RUJ_READS_START	INTEGER	
RUJ_WRITES_START	INTEGER	
AIJ_WRITES_START	INTEGER	
ROOT_READS_START	INTEGER	
ROOT_WRITES_START	INTEGER	
BUFFER_READS_START	INTEGER	
GET_VM_BYTES_START	INTEGER	
FREE_VM_BYTES_START	INTEGER	
LOCK_REQS_START	INTEGER	
REQ_NOT_QUEUED_START	INTEGER	
REQ_STALLS_START	INTEGER	
REQ_DEADLOCKS_START	INTEGER	
PROM_DEADLOCKS_START	INTEGER	
LOCK_RELS_START	INTEGER	

(continued on next page)

Table 6–5 (Cont.) Columns for Table EPC$1_221_REQUEST_ACTUAL

Column Name	Data Type	Domain
LOCK_STALL_TIME_START	INTEGER	
D_FETCH_RET_START	INTEGER	
D_FETCH_UPD_START	INTEGER	
D_LB_ALLOK_START	INTEGER	
D_LB_GBNEEDLOCK_START	INTEGER	
D_LB_NEEDLOCK_START	INTEGER	
D_LB_OLDVER_START	INTEGER	
D_GB_NEEDLOCK_START	INTEGER	
D_GB_OLDVER_START	INTEGER	
D_NOTFOUND_IO_START	INTEGER	
D_NOTFOUND_SYN_START	INTEGER	
S_FETCH_RET_START	INTEGER	
S_FETCH_UPD_START	INTEGER	
S_LB_ALLOK_START	INTEGER	
S_LB_GBNEEDLOCK_START	INTEGER	
S_LB_NEEDLOCK_START	INTEGER	
S_LB_OLDVER_START	INTEGER	
S_GB_NEEDLOCK_START	INTEGER	
S_GB_OLDVER_START	INTEGER	
S_NOTFOUND_IO_START	INTEGER	
S_NOTFOUND_SYN_START	INTEGER	
D_ASYNC_FETCH_START	INTEGER	
S_ASYNC_FETCH_START	INTEGER	
D_ASYNC_READIO_START	INTEGER	
S_ASYNC_READIO_START	INTEGER	
AS_READ_STALL_START	INTEGER	
AS_BATCH_WRITE_START	INTEGER	
AS_WRITE_STALL_START	INTEGER	
BIO_START	INTEGER	
DIO_START	INTEGER	
PAGEFAULTS_START	INTEGER	
PAGEFAULT_IO_START	INTEGER	
CPU_START	INTEGER	
CURRENT_PRIO_START	SMALLINT	
VIRTUAL_SIZE_START	INTEGER	
WS_SIZE_START	INTEGER	
WS_PRIVATE_START	INTEGER	
WS_GLOBAL_START	INTEGER	

(continued on next page)

Table 6–5 (Cont.) Columns for Table EPC$1_221_REQUEST_ACTUAL

Column Name	Data Type	Domain
CLIENT_PC_END	INTEGER	
STREAM_ID_END	INTEGER	
REQ_ID_END	INTEGER	
COMP_STATUS_END	INTEGER	
REQUEST_OPER_END	INTEGER	
TRANS_ID_END	VARCHAR(16)	
TRANS_ID_END_STR_ID	INTEGER	STR_ID_DOMAIN
DBS_READS_END	INTEGER	
DBS_WRITES_END	INTEGER	
RUJ_READS_END	INTEGER	
RUJ_WRITES_END	INTEGER	
AIJ_WRITES_END	INTEGER	
ROOT_READS_END	INTEGER	
ROOT_WRITES_END	INTEGER	
BUFFER_READS_END	INTEGER	
GET_VM_BYTES_END	INTEGER	
FREE_VM_BYTES_END	INTEGER	
LOCK_REQS_END	INTEGER	
REQ_NOT_QUEUED_END	INTEGER	
REQ_STALLS_END	INTEGER	
REQ_DEADLOCKS_END	INTEGER	
PROM_DEADLOCKS_END	INTEGER	
LOCK_RELS_END	INTEGER	
LOCK_STALL_TIME_END	INTEGER	
D_FETCH_RET_END	INTEGER	
D_FETCH_UPD_END	INTEGER	
D_LB_ALLOK_END	INTEGER	
D_LB_GBNEEDLOCK_END	INTEGER	
D_LB_NEEDLOCK_END	INTEGER	
D_LB_OLDVER_END	INTEGER	
D_GB_NEEDLOCK_END	INTEGER	
D_GB_OLDVER_END	INTEGER	
D_NOTFOUND_IO_END	INTEGER	
D_NOTFOUND_SYN_END	INTEGER	
S_FETCH_RET_END	INTEGER	
S_FETCH_UPD_END	INTEGER	
S_LB_ALLOK_END	INTEGER	
S_LB_GBNEEDLOCK_END	INTEGER	

(continued on next page)

Table 6–5 (Cont.) Columns for Table EPC$1_221_REQUEST_ACTUAL

Column Name	Data Type	Domain
S_LB_NEEDLOCK_END	INTEGER	
S_LB_OLDVER_END	INTEGER	
S_GB_NEEDLOCK_END	INTEGER	
S_GB_OLDVER_END	INTEGER	
S_NOTFOUND_IO_END	INTEGER	
S_NOTFOUND_SYN_END	INTEGER	
D_ASYNC_FETCH_END	INTEGER	
S_ASYNC_FETCH_END	INTEGER	
D_ASYNC_READIO_END	INTEGER	
S_ASYNC_READIO_END	INTEGER	
AS_READ_STALL_END	INTEGER	
AS_BATCH_WRITE_END	INTEGER	
AS_WRITE_STALL_END	INTEGER	
BIO_END	INTEGER	
DIO_END	INTEGER	
PAGEFAULTS_END	INTEGER	
PAGEFAULT_IO_END	INTEGER	
CPU_END	INTEGER	
CURRENT_PRIO_END	SMALLINT	
VIRTUAL_SIZE_END	INTEGER	
WS_SIZE_END	INTEGER	
WS_PRIVATE_END	INTEGER	
WS_GLOBAL_END	INTEGER	

Table 6–6 shows the TRANSACTION table.

Table 6–6 Columns for Table EPC$1_221_TRANSACTION

Column Name	Data Type	Domain
COLLECTION_RECORD_ID	SMALLINT	COLLECTION_RECORD_ID_DOMAIN
IMAGE_RECORD_ID	INTEGER	IMAGE_RECORD_ID_DOMAIN
CONTEXT_NUMBER	INTEGER	CONTEXT_NUMBER_DOMAIN
TIMESTAMP_START	DATE VMS	
TIMESTAMP_END	DATE VMS	
CLIENT_PC_START	INTEGER	
STREAM_ID_START	INTEGER	
LOCK_MODE_START	INTEGER	
TRANS_ID_START	VARCHAR(16)	

(continued on next page)

Table 6–6 (Cont.) Columns for Table EPC$1_221_TRANSACTION

Column Name	Data Type	Domain
TRANS_ID_START_STR_ID	INTEGER	STR_ID_DOMAIN
GLOBAL_TID_START	VARCHAR(16)	
GLOBAL_TID_START_STR_ID	INTEGER	STR_ID_DOMAIN
DBS_READS_START	INTEGER	
DBS_WRITES_START	INTEGER	
RUJ_READS_START	INTEGER	
RUJ_WRITES_START	INTEGER	
AIJ_WRITES_START	INTEGER	
ROOT_READS_START	INTEGER	
ROOT_WRITES_START	INTEGER	
BUFFER_READS_START	INTEGER	
GET_VM_BYTES_START	INTEGER	
FREE_VM_BYTES_START	INTEGER	
LOCK_REQS_START	INTEGER	
REQ_NOT_QUEUED_START	INTEGER	
REQ_STALLS_START	INTEGER	
REQ_DEADLOCKS_START	INTEGER	
PROM_DEADLOCKS_START	INTEGER	
LOCK_RELS_START	INTEGER	
LOCK_STALL_TIME_START	INTEGER	
D_FETCH_RET_START	INTEGER	
D_FETCH_UPD_START	INTEGER	
D_LB_ALLOK_START	INTEGER	
D_LB_GBNEEDLOCK_START	INTEGER	
D_LB_NEEDLOCK_START	INTEGER	
D_LB_OLDVER_START	INTEGER	
D_GB_NEEDLOCK_START	INTEGER	
D_GB_OLDVER_START	INTEGER	
D_NOTFOUND_IO_START	INTEGER	
D_NOTFOUND_SYN_START	INTEGER	
S_FETCH_RET_START	INTEGER	
S_FETCH_UPD_START	INTEGER	
S_LB_ALLOK_START	INTEGER	
S_LB_GBNEEDLOCK_START	INTEGER	
S_LB_NEEDLOCK_START	INTEGER	
S_LB_OLDVER_START	INTEGER	
S_GB_NEEDLOCK_START	INTEGER	
S_GB_OLDVER_START	INTEGER	

(continued on next page)

Table 6–6 (Cont.) Columns for Table EPC$1_221_TRANSACTION

Column Name	Data Type	Domain
S_NOTFOUND_IO_START	INTEGER	
S_NOTFOUND_SYN_START	INTEGER	
D_ASYNC_FETCH_START	INTEGER	
S_ASYNC_FETCH_START	INTEGER	
D_ASYNC_READIO_START	INTEGER	
S_ASYNC_READIO_START	INTEGER	
AS_READ_STALL_START	INTEGER	
AS_BATCH_WRITE_START	INTEGER	
AS_WRITE_STALL_START	INTEGER	
AREA_ITEMS_START	VARCHAR(128)	
AREA_ITEMS_START_STR_ID	INTEGER	STR_ID_DOMAIN
BIO_START	INTEGER	
DIO_START	INTEGER	
PAGEFAULTS_START	INTEGER	
PAGEFAULT_IO_START	INTEGER	
CPU_START	INTEGER	
CURRENT_PRIO_START	SMALLINT	
VIRTUAL_SIZE_START	INTEGER	
WS_SIZE_START	INTEGER	
WS_PRIVATE_START	INTEGER	
WS_GLOBAL_START	INTEGER	
CROSS_FAC_2_START	INTEGER	
CROSS_FAC_3_START	INTEGER	
CROSS_FAC_7_START	INTEGER	
CROSS_FAC_14_START	INTEGER	
DBS_READS_END	INTEGER	
DBS_WRITES_END	INTEGER	
RUJ_READS_END	INTEGER	
RUJ_WRITES_END	INTEGER	
AIJ_WRITES_END	INTEGER	
ROOT_READS_END	INTEGER	
ROOT_WRITES_END	INTEGER	
BUFFER_READS_END	INTEGER	
GET_VM_BYTES_END	INTEGER	
FREE_VM_BYTES_END	INTEGER	
LOCK_REQS_END	INTEGER	
REQ_NOT_QUEUED_END	INTEGER	
REQ_STALLS_END	INTEGER	

(continued on next page)

Table 6–6 (Cont.) Columns for Table EPC$1_221_TRANSACTION

Column Name	Data Type	Domain
REQ_DEADLOCKS_END	INTEGER	
PROM_DEADLOCKS_END	INTEGER	
LOCK_RELS_END	INTEGER	
LOCK_STALL_TIME_END	INTEGER	
D_FETCH_RET_END	INTEGER	
D_FETCH_UPD_END	INTEGER	
D_LB_ALLOK_END	INTEGER	
D_LB_GBNEEDLOCK_END	INTEGER	
D_LB_NEEDLOCK_END	INTEGER	
D_LB_OLDVER_END	INTEGER	
D_GB_NEEDLOCK_END	INTEGER	
D_GB_OLDVER_END	INTEGER	
D_NOTFOUND_IO_END	INTEGER	
D_NOTFOUND_SYN_END	INTEGER	
S_FETCH_RET_END	INTEGER	
S_FETCH_UPD_END	INTEGER	
S_LB_ALLOK_END	INTEGER	
S_LB_GBNEEDLOCK_END	INTEGER	
S_LB_NEEDLOCK_END	INTEGER	
S_LB_OLDVER_END	INTEGER	
S_GB_NEEDLOCK_END	INTEGER	
S_GB_OLDVER_END	INTEGER	
S_NOTFOUND_IO_END	INTEGER	
S_NOTFOUND_SYN_END	INTEGER	
D_ASYNC_FETCH_END	INTEGER	
S_ASYNC_FETCH_END	INTEGER	
D_ASYNC_READIO_END	INTEGER	
S_ASYNC_READIO_END	INTEGER	
AS_READ_STALL_END	INTEGER	
AS_BATCH_WRITE_END	INTEGER	
AS_WRITE_STALL_END	INTEGER	
AREA_ITEMS_END	VARCHAR(128)	
AREA_ITEMS_END_STR_ID	INTEGER	STR_ID_DOMAIN
BIO_END	INTEGER	
DIO_END	INTEGER	
PAGEFAULTS_END	INTEGER	
PAGEFAULT_IO_END	INTEGER	
CPU_END	INTEGER	

(continued on next page)

Table 6–6 (Cont.) Columns for Table EPC$1_221_TRANSACTION

Column Name	Data Type	Domain
CURRENT_PRIO_END	SMALLINT	
VIRTUAL_SIZE_END	INTEGER	
WS_SIZE_END	INTEGER	
WS_PRIVATE_END	INTEGER	
WS_GLOBAL_END	INTEGER	
CROSS_FAC_2_END	INTEGER	
CROSS_FAC_3_END	INTEGER	
CROSS_FAC_7_END	INTEGER	
CROSS_FAC_14_END	INTEGER	

Table 6–7 shows the REQUEST_BLR table.

Table 6–7 Columns for Table EPC$1_221_REQUEST_BLR

Column Name	Data Type	Domain
COLLECTION_RECORD_ID	SMALLINT	COLLECTION_RECORD_ID_ DOMAIN
IMAGE_RECORD_ID	INTEGER	IMAGE_RECORD_ID_DOMAIN
CONTEXT_NUMBER	INTEGER	CONTEXT_NUMBER_DOMAIN
TIMESTAMP_POINT	DATE VMS	
CLIENT_PC	INTEGER	
STREAM_ID	INTEGER	
REQ_ID	INTEGER	
TRANS_ID	VARCHAR(16)	
TRANS_ID_STR_ID	INTEGER	STR_ID_DOMAIN
REQUEST_NAME	VARCHAR(31)	
REQUEST_NAME_STR_ID	INTEGER	STR_ID_DOMAIN
REQUEST_TYPE	INTEGER	
BLR	VARCHAR(127)	
BLR_STR_ID	INTEGER	STR_ID_DOMAIN

7

Known Problems and Restrictions

This chapter describes problems and restrictions relating to Oracle Trace and includes workarounds where appropriate.

7.1 Known Problems and Restrictions

This section describes known problems and restrictions that affect all interfaces. This is not an exhaustive list.

7.1.1 Oracle Trace Data not Being Collected for all Database Connections

Bug 9728436

Releases of Oracle Rdb Server prior to release 7.3.3.1 failed to record Oracle Trace workloads for a second and subsequent database attaches; via the ATTACH, DECLARE ALIAS or CONNECT statements. Only workload performance statistics were collected for the initial database attach.

This problem has been corrected in Oracle Rdb Server release 7.3.3.1 and later versions. Oracle Rdb Server now correctly initializes the Oracle Trace interface for each attach during a session; now complete workloads are collected.

7.1.2 NULL Elimination Semantics now Supported by COUNT Function

In prior versions of Oracle Rdb Server the COUNT aggregate function did not support the ANSI and ISO SQL Language Standard NULL Elimination semantics. This support should report the warning *null value eliminated in set function* whenever the row set included a NULL value that was ignored when computing the count of the value-expression. This problem affected both COUNT(value_expression) and COUNT(DISTINCT value_expression) variants of the COUNT aggregate function.

In prior versions these functions were implicitly translated by SQL to COUNT (*) FILTER (WHERE value_expression IS NOT NULL) and therefore NULL values were never processed by the COUNT function. This can be seen in the strategy display shown in the following example.

```
SQL> select count(middle_initial)
cont> from employees;
Tables:
  0 = EMPLOYEES
Aggregate: 0:COUNT (0.MIDDLE_INITIAL) Q2
             Bool: NOT MISSING (0.MIDDLE_INITIAL)
Get     Retrieval sequentially of relation 0:EMPLOYEES

          64
1 row selected
SQL>
```

This problem has been corrected in Oracle Rdb Server release 7.3.1 and later versions. Please note that query encoding for COUNT (value_expression) and COUNT (DISTINCT value_expression) was changed in that release and any query outlines defined for queries, views or procedures using these functions would need to be recreated.

This change in Oracle Rdb Server caused the problem reported in Section 2.1.2.