Oracle Trace for OpenVMS

Table of Contents

Oracle Trace for OpenVMS

Release Notes

Release 7.2.0.3

June 2010

This document provides release notes for Oracle Trace for OpenVMS for the HP OpenVMS operating system.

Oracle Trace for OpenVMS Release Notes Release 7.2.0.3 for the operating system.

quality of third−party products or services, or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Chapter 1
Release Notes for Oracle Trace for OpenVMS Release 7.2.0.3

This document provides release notes for Oracle Trace for the HP OpenVMS operating system. The release notes describe restrictions and problems that have been corrected for this version of Oracle Trace for OpenVMS that are not documented elsewhere.

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:

```
$ COLL STOP SYSTEM/ABORT
```

Install the kit using the VMSINSTAL utility by logging in to the SYSTEM account and typing the following command for an Itanium system:

```
$ SYS$UPDATE:VMSINSTAL EPC07203I072 [saveset location]
```

Install the kit using the VMSINSTAL utility by logging in to the SYSTEM account and typing the following command for an Alpha system:

```
$ SYS$UPDATE:VMSINSTAL EPC07203A072 [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 7.2 for complete installation instructions.

1.2 Minimum VMS and Rdb Versions

This release of Oracle Trace requires the following software:

- OpenVMS Alpha V8.2 or later
- OpenVMS IA64 V8.2−1 or later
- Oracle Rdb V7.2−0 or later for EPC databases

1.3 Software Errors Fixed in Oracle Trace Release 7.2.0.3

1.3.1 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.

Note that a larger Data Collection file will require more disk space for the Data Collection file, the Rdb database or RMS file created by the COLLECT FORMAT command, and report files created by the COLLECT REPORT command. 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 COLLECT SCHEDULE COLLECTION "/COLLECTION_FILES=(MAX_ALLOCATION=blocks)" command qualifier. The actual attainable maximum block size is also limited by available disk space and any VMS 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 a Oracle Trace collection.

This problem has been corrected in Oracle Trace release 7.2.0.3.

1.3.2 Oracle Trace Collections Could Select the Wrong Facility or Facility Version

There was a problem in Oracle Trace V7 versions where, if there was more than one facility selection defined in the Oracle Trace Administration database by the COLLECT CREATE SELECTION command, facility selection definitions which were not specified in the COLLECT SCHEDULE COLLECTION command could be chosen for Oracle Trace collections. This could result in no data being collected since the wrong facility version was used for the collection, or data being collected from other facilities or other facility versions if those facilities were being executed at the time when the Oracle Trace collection was active. If only one facility selection was defined in the Oracle Trace Administration database when the Oracle Trace Collection was scheduled this problem would not happen. This problem has been fixed and now no matter how many facility selection definitions are defined in the Oracle Trace Administration database only the selection specified in the COLLECT SCHEDULE COLLECTION command will be used for the collection.

The following example shows the problem. Even though the selection RDB_SELECTION is specified for the RDB_COLLECTION collection the selections ALL_RDB and DBMS_ALL are used to select facility data for the RDB_COLLECTION collection. As a result, no event data is collected when the RDB_COLLECTION collection is run. When the Oracle Trace Data Collection file RDB_SELECTION.DAT is formated to the RDB_SELECTION.RDB database, the EPC$EVENT table is empty and the EPC$FACILITY table shows that the ALL_RDB and DBMS_ALL selections were used to select data for the RDB_COLLECTION collection, not the RDB_SELECTION selection. Since the RDBVMS T7.2 facility and the DBMS X7.2

facility were not running at the time of the collection, no event data was collected.

```
$ COLLECT SHOW SELECTION 4-NOV-2009 10:55         Facility Selection Information              Pag
```

To avoid this problem have only one facility selection defined in the Oracle Trace Administration database when executing the COLLECT SHEDULE COLLECTION command.

This problem has been corrected in Oracle Trace release 7.2.0.3.

1.3.3 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 VAR (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

1.3.4 Unexpected Looping When Using the BLR Converter

Bug 8881681

It was possible that 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 error in future.

1.3.5 Various Memory Leaks Corrected

When using various report generation and analysis functions, a process could unexpected consume all of of P0 virtual address space and fail with a variety of 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 somewhat reduced and performance improved in some cases.

1.3.6 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 whiles formatting the data collection file.

This problem has been corrected in Oracle TRACE Release 7.2.0.3.

1.3.7 File Size Limit on Oracle Trace 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.

1.4 Software Errors Fixed in Oracle Trace Release 7.2.0.2

1.4.1 Oracle Trace %SQL−F−NOTXNOUT Error if FORMAT/ELAPSED_TIME7202

There was a problem in Oracle Trace where a %SQL−F−NOTXNOUT error was output and a dump file was generated when the ELAPSED_TIME qualifier was specified with the FORMAT command. For the Oracle Trace FORMAT command, used to format Oracle Trace Data Collection files, the ELAPSED_TIME qualifier can be used to calculate the elapsed time for each duration event and then store the result in the Oracle RDB database which the FORMAT command creates and stores the formatted data to. NOELAPSED_TIME is the default. ELPASED_TIME is only for formatting to RDB databases, not RMS files. This qualifier is only needed by users intending to generate their own reports from the single file Oracle Rdb database which the FORMAT command creates. If instead the Oracle Trace Report Generator is used to create reports it calculates the elapsed time automatically.

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 in the collection file to the database tried to commit a database transaction without having previously started one. This caused the format operation to abort with the %SQL−F−NO_TXNOUT error and the dump. This problem has been fixed and now the format operation will succeed and return the %EPC−S−FMT_DCF_SUCCESS status.

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 dat
```

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 dat
```

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

This problem has been corrected in Oracle Trace release 7.2.0.2.

1.5 Software Errors Fixed in Oracle Trace Release 7.2.0.1

1.5.1 Invalid Oracle Trace Summary Report Elapsed Time Statistics

There was a problem with the Summary Report Elapsed Time statistics output by Oracle Trace on the HP Open VMS Integrity platform which caused incorrect Elapsed Time values to be put out. These values could consist of large positive or large negative numbers or the value "NaNQ". This happened because Oracle Trace on the Integrity platform 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 on the Integrity platform.

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. The Elapsed Time statistics could also contain the value "NaNQ" instead of a numerical value.

```
$ SET VERIFY $ @REPORTRDB.COM COLLECT REPORT RDB_DB- /TYPE=SUMMARY- /TITLE="RDB/VMS SUMMARY REPOR
```

This problem has been corrected in Oracle Trace Release 7.2.0.1.

1.5.2 Oracle Trace 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 the HP Open VMS Integrity platform 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 precison 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. Note that this problem only happened for Summary Report Standard Deviation and 95 Percentile statistics on the Integrity platform.

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 REPOR
```

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.

1.6 Software Errors Fixed in Trace Release 7.2

This kit contains all fixes made to previous versions of Oracle Trace for OpenVMS and also addresses the problems described in the following sections.

1.6.1 ORACLE Trace Did Not Collect Data for the RDBVMS Facility

Oracle Trace did not collect data for the RDBVMS facility for Oracle Rdb Release 7.1−100 and Release 7.1−101. This was because the version number specified by Oracle Rdb to the Oracle Trace Service Routine EPC$INIT was V7.1−10, but the version number inserted by the Oracle Rdb installation into the module RDBVMSV7.1−10 in the Oracle Trace VMS SYS$SHARE:EPC$FACILITY.TLB text library was V7.1−100 or V7.1−101. This module could then be extracted and used to set the RDBVMS facility definition and version in the Oracle Trace Administration database. The version string passed to the Oracle Trace Service Routine EPC$INIT must match identically the version specified in the facility definition in the Oracle Trace Administration database. Because this was not true for the RDBVMS facility, when an RDBVMS Oracle Trace collection was scheduled, no RDBVMS data was collected.

This has been corrected for Oracle Rdb Release 7.1−200 and later. The version number passed to EPC$INIT for the RDBVMS facility will exactly match the RDBVMS facility version inserted by the Oracle Rdb installation procedure into SYS$SHARE:EPC$FACILITY.TLB, which can then be used to define the RDBVMS facility in the Oracle Trace Administration database. For all Oracle Rdb 7.2−n releases, the version is V7.2.

The following example shows that for Oracle Rdb Release 7.1−100 and Release 7.1−101, the RDBVMS facility data was not getting collected (no arrow points to the RDBVMS facility) when the Oracle Trace COLLECT SHOW REGISTER/NOCLUSTER command is issued. Data is getting collected for the ATM_SAMPLE facility as indicated by the arrow.

```
$ COLLECT SHOW REGISTER/NOCLUSTER  Registrations actively collecting    Node: CLNODE  Collection:
```

The only workaround for this problem would be to change the version numbers in the Oracle Trace administration database tables from V7.100 or V7.101 to V7.1−10.

1.6.2 Finding the Transaction Type of a Particular Transaction within a Oracle Trace Database

The table EPC$1_221_TRANSACTION in the formatted Oracle Trace database has a column named LOCK_MODE_START of longword datatype. The values of this column indicate the type of transaction a particular transaction was:

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

1.6.3 Oracle Trace Collect Format Command Could Not Create a Database After Rdb Upgrade

There was a problem with Oracle Trace V2.4.1 where, after an upgrade to Oracle Rdb Release 7.0.7.1, the Collect Format command was no longer able to successfully create the database where the formatted data is stored. This problem was fixed in Oracle Trace V2.4.1.1 Update 01. Note that this problem has been fixed in Oracle Trace, not Oracle Rdb.

The following example shows the problem that the Oracle Trace Collect Format command had creating a database when Oracle Rdb was upgraded to Oracle Rdb Release 7.0.7.1.

```
$ COLLECT FORMAT SAMPLE_DATA.DAT SAMPLE_DATA.RDB %EPC-E-FMT_FAILURE, Formatting failed -RDMS-F-NO
```

| Contents