How To Trace From Form, Report, Program And Others In Oracle Applications (Doc ID 130182.1)

To Bottom

APPLIES TO:

Oracle Application Object Library - Version 11.5.10.0 to 12.1.1 [Release 11.5 to 12.1]
Information in this document applies to any platform.

PURPOSE

The purpose of this note is to bring together methods to activate and retrieve trace when executing forms, reports, executables, and others involved in Oracle Applications v. 11.x.

SCOPE

All audience. Trace files are often needed for support purpose. Customers don't always know how to activate trace and retrieve trace file.

DETAILS

CONTENTS
----

1. TRACING DATABASE INSTANCE

Tracing the entire database is not the best way to analyze an issue in Oracle Applications but it could be the only means in particular cases.
Principle is to :
- Edit init<SID>.ora file
- Add 'SQL_TRACE=TRUE' or 'EVENT=...' parameters
- Verify presence of 'TIMED_STATISTICS=TRUE' parameter to track
and analyze performance issues
- Stop concurrent managers and shutdown database
- Restart database and concurrent managers

2. TRACING A SESSION

You could need to trace a particular session for example if issue occurs for only one user or if you can't shutdown database.
Principle is to :
- Retrieve number from a current session of Oracle Applications,
the easiest way is to :
- Connect to System Administrator responsibility
- Menu Security => User => Monitor
- Note the Oracle Process (PID in fact) for the proper User Name
- Connect to sqlplus (system account)
- Run this script to get session and serial numbers :

```
select p.pid, p.spid, s.sid, s.serial#, s.username
from v$process p , v$session s
where p.addr = s.paddr and p.pid = &your_pid;
```

=> enter PID found before it will give you needed session and serial numbers

- now run this command to activate trace for your session :

```
execute dbms_system.set_sql_trace_in_session(<SID>, <SERIAL#>, TRUE)
```

See Note:115675.1 for other methods.

3. TRACING FROM A FORM

You can activate/deactivate database trace from any screen in Oracle Applications via menu Help => Tools => Trace.

4. TRACING A FORM

You can also want to trace the screen itself to collect information on triggers or built-ins which are executed, or capture messages and unhandled exceptions...
Principle is to :
- edit html file which launch Oracle Applications
- complete <PARAM name="serverArgs" value="....."> with two values :
record=collect and log=<your file name>

This method is available from 10.7NCA to rel 11.x.
See Note:1020627.102 for more details.

A debug utility exists for some AR screens :
Transactions (ARXTWMAI)
Receipts (ARXRWMAI)
Collections (ARXCWMAI)
Principle is to add parameter AR_DEBUG_FLAG via Option Examine

5. TRACING A REPORT

Beginning rel 11.x that becomes easy to activate database trace for a single report. You just need to check trace option box in concurrent program definition (under System Administrator responsability).

If trace option is not available or you report is customized you can insert an 'alter session set sql_trace=TRUE' in it to activate database trace.

Principle is to :
- make a copy of your report
- generate a .rex file from .rdf (by command r25convm or r25conv32, see respectively Note:1070541.6 and Note:1020489.102)
- edit the .rex file and insert the line immediately after ('FND SRWINIT'); :
srw.do_sql('alter session set sql_trace=TRUE');
- insert another line immediately after ('FND SRWEXIT'); to stop tracing :
srw.do_sql('alter session set sql_trace=FALSE');
- regenerate the .rdf file

See excellent Note:1019231.6 for complete procedure.

6. TRACING AX

For example it is interesting to trace Posting Manager when issue occurs.
Principle is to set profile options:
AX: Debug Mode to Yes
AX: Debug Level from 1 to 100

You will find the output in the concurrent request log.

7. TRACING OTHER CONCURRENT PROGRAMS

As for report you can check trace option box for any concurrent program. There exist also profile options to activate trace when particular concurrent programs are running.
For instance you can set these profile options below at user level :

INV: Debug Trace
MRP: Trace Mode
OE: Debug Trace
...

8. TRACING A PROCESS

It is sometimes interesting to retrieve running sql statement when a program hangs and process is pending.
Principle is to :
- retrieve process id (see Note.105395.1)
- run svrmgrl and connect with system account
- execute command below :
oradebug setospid <process id>
- activate tracing by :
oradebug event 10046 trace name context forever, level 12
- deactivate tracing by :
oradebug event 10046 trace name context off

See Note:76338.1 for more details and other possibilities.

9. RETRIEVING TRACE FILE

Directory where trace files reside is defined by variable USER_DUMP_DEST, if it is not set, execute sql command below from sqlplus (apps account):

select value from v$parameter
where name='user_dump_dest';

or execute command below under svrmgrl :

show parameters user_dump_dest

Trace files are named ora_<SPID>.trc. To retrieve your trace file you can try unix command :

ls -lt
verify timestamp and also search a characteristic string with grep command.

A more precise method consists to find PID (Oracle Process Id) and determine corresponding SPID (Session Process Id):

- under System Administrator responsibility,
menu Security => User => Monitor
column 'Oracle' (bloc 'SIGNON_AUDIT_VIEW', field 'PID') gives you the PID
corresponding to your user name.
- under sqlplus, apps or system account, run :

select p.pid, p.spid
from v$process p , v$session s
where p.addr = s.paddr and p.pid = &your_pid;

=> enter PID found before it will give you SPID
- now you can retrieve precisely ora_<SPID>.trc file.

10. FORMATING TRACE FILE

Use tkprof utility to format the raw trace in more readable file.
For example run this command from unix :

tkprof <raw trace file .trc> <any output name> explain=apps/<apps password>

REFERENCES

NOTE:1020627.102 - NEW:HOW TO RUN A LEVEL 4 FORMS TRACE USING D2K
NOTE:115675.1 - How to Trace Sessions Using Event 10046 With Level xx in Oracle Applications
NOTE:117129.1 - How to get a Trace first And Begin to Analyze a E-Business Applications Performance Issue
NOTE:117820.1 - Troubleshooting (Tracing)
NOTE:47837.1 - Applications Utilities FAQ
NOTE:62664.1 - Forms Server Logging and Forms Runtime Diagnostics (FRD) Explained
NOTE:76338.1 - Tracing Tips for Oracle Applications

Related

Products

- Oracle E-Business Suite > Applications Technology > Application Object Library > Oracle Application Object Library > Basic SysAdmin functions, maintenance

Keywords

SQL_TRACE; TRACE; TRACING OPTIONS; USER_DUMP_DEST