Tracing and Debugging
RDC/TMS/OC: The Secrets of the Oracle RDBMS and OPA Debugging Tools in the OPA 4.0 Environment

DBMS Consulting, Inc. Sunil G. Singh
Introduction

- Sunil G. Singh of DBMS Consulting, Inc.
- Specialize in Oracle Pharmaceutical and E-Business implementations and long-term support.
- Thanks to the OCUG for this demo/discussion opportunity.
Acknowledgements

- Thanks to OCUG for this opportunity to speak.
- Special thanks to my many friends and colleagues at Oracle, OCS, OPA Support, Industry and other Vendors for their comments and insights into this topic.
- Special thanks to S. Clover, K. Howells, A. Alasso and K. Rejndrup for some of the OPA specific techniques discussed here.
Goals

- Explanation of the needs for special debugging techniques for RDC, TMS and OC from a customer perspective and some possible issues.
- Examine some general debugging techniques from the Oracle RDBMS, pertaining to OPA in a production environment.
- Examine some general debugging techniques from the UNIX OS level.
- Examine some general debugging techniques techniques that are OPA specific related to Oracle Forms and Server Code.
Scope

- Technical discussion.
- Debugging techniques considered as related to the Oracle Pharmaceutical Applications only.
- Debugging from a developer’s perspective is not covered. No source code access is assumed.
- Absolutely no intention or attempt to reverse engineer any Oracle product or violate any Oracle license agreement. The content of this presentation is only for production support purposes.
Assumptions

• OC 4.0.2 or OC 4.0.3 configuration with a UNIX back-end.
• Privileged access to NT Middle Tier and RDBMS back-end.
What environments should be used

- Where possible, all debugging should occur in a testing or development environment.
- An instance which is a copy of the production environment is also an invaluable asset for issues which are data or volume related.
- If some debugging or analysis of a problem must be done in a production environment, great care must be taken to insure that it does not greatly impact production work or affect production data.
Reasons for Debugging in a production OPA environment

• Occasionally, an error may be encountered in an Oracle Pharmaceutical Application which is not possible to reproduce in a test or development environment. It is useful in these cases to have analytical techniques available in production.

• To expedite the resolution and analysis process, it is always necessary to:
  – Determine if the problem is unique to a specific environment, server, instance, Middle Tier, study, user, PC, OS.
  – Determine if the problem is the result of something unique to a specific organisation’s configuration.
  – Determine if the problem is related to a customisation.
Who should Debug? Access and privileges to these techniques.

- Roles for administrative support for OC within an organisation need to be clearly defined.
- It is very difficult to segregate the role of application support completely from DBA or System Administration support. In an ideal situation, the resources responsible for OPA support would have complete access to the OS and RDBMS as required by these techniques.
- However, this may contradict already defined roles for User, Application and System support.
Assume a production system where RDBMS can not be shutdown

Created a test scenario where the error “32200: Unable to set printer specification” appears when selecting Admin -> Admin Reports -> Reference Codelists

Goal is to trap the exact SQL statement that causes the error so that the root cause can be determined

The key is to start the trace right before the error actually occurs. Don’t trace at the login time of the user because the trace file will be too large and contain superfluous SQL statements.
RDBMS Level: dbms_system (2)

- Bring application right to the brink of the error/failure.
- Identify the sid, serial#, username from v$session
  - Select sid, serial#, username, program from v$session where username = 'OPS$<user>';
- Start the tracing:
  - Execute
    ```sql
    sys.dbms_system.set_sql_trace_in_session(<SID>,<SERIAL#>,TRUE);
    ```
- Turn on bind variables
  - Execute
    ```sql
    sys.dbms_system.set_ev(<SID>,<SERIAL#>,10046,12,'''');
    ```
RDBMS Level: dbms_system (3)

- Actually cause the error to occur. Perform the action that causes the failure
- Do NOT close the dialogue box
- Stop the tracing:
  - Execute `sys.dbms_system.set_sql_trace_in_session(<SID>,<SERIAL#>,FALSE);`
- Turn off bind variables
  - Execute `sys.dbms_system.set_ev(<SID>,<SERIAL#>,10046,0,'');`
- Copy the trace file into a temporary directory. Check the USER_DUMP_DEST area for a current trace file.
RDBMS Level: dbms_system (4)

- Execute tkprof and generate a record file
  - Tkprof <trace_file>.trc <trace_file>.out record=<trace_file>.rec
- Examine the record file and find the SQL Statement before the MESSAGE_TOPICS select
- Find this statement in the .trc file. Examine the value= line in this file to determine the value of the bind variable, e.g., :b1
- Execute this SQL statement interactively to determine the root cause of the failure.
Oracle Server Manager Release 3.1.7.0.8 - Production

Copyright (c) 1997, 1999, Oracle Corporation. All Rights Reserved.

Oracle8i Enterprise Edition Release 8.1.7.2.0 - Production
With the Partitioning option
JServ Release 8.1.7.2.0 - Production

SVRMGR> connect internal
Connected.
SVRMGR> select sid, serial#, username, program from v$session where username = 'OPS$OPAPP$';

<table>
<thead>
<tr>
<th>SID</th>
<th>SERIAL#</th>
<th>USERNAME</th>
<th>PROGRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>2567</td>
<td>OPS$OPAPP$</td>
<td></td>
</tr>
</tbody>
</table>

1 row selected.
SVRMGR> execute sys.dbms_system.set_sql_trace_in_session(15,2567,TRUE);
Statement processed.
SVRMGR> execute sys.dbms_system.set_ev(15,2567,10046,12,'');
Statement processed.
OCUG San Diego 2002: OPA Debugging Techniques in a 4.0 Environment

```
SVRMRG> execute sys.dbms_system.set_sql_trace_in_session(15,2567,FALSE);
Statement processed.
SVRMRG> execute sys.dbms_system.set_sql_trace_in_session(15,2567,FALSE);
Statement processed.
SVRMRG> execute sys.dbms_system.set_ev(15,2567,10046,0,'');
Statement processed.
SVRMRG> show parameter dump

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>background_core_dump</td>
<td>string</td>
<td>partial</td>
</tr>
<tr>
<td>background_dump_dest</td>
<td>string</td>
<td>/export/home/oracle/admin/octm</td>
</tr>
<tr>
<td>core_dump_dest</td>
<td>string</td>
<td>/export/home/oracle/admin/octm</td>
</tr>
<tr>
<td>max_dump_file_size</td>
<td>string</td>
<td>UNLIMITED</td>
</tr>
<tr>
<td>shadow_core_dump</td>
<td>string</td>
<td>partial</td>
</tr>
<tr>
<td>user_dump_dest</td>
<td>string</td>
<td>/export/home/oracle/admin/octm</td>
</tr>
</tbody>
</table>

SVRMRG> exit
Server Manager complete.
```

dbmssunserver3% cd /export/home/oracle/admin/octms403/udump
dbmssunserver3% ls -alt | more
total 194
```
drwxr-xr-x 2 oracle dba 1536 Sep 24 06:36 .
-rw-r----- 1 oracle dba 51687 Sep 24 06:36 octms403_ora_893.trc
dbmssunserver3% cp octms403_ora_893.trc /tmp/ref_codelist_error.trc
dbmssunserver3% cd /tmp
dbmssunserver3% tkprof ref_codelist_error.trc ref_codelist_error.out record=ref_codelist_error.rec
```

TKPROF: Release 8.1.7.2.0 - Production on Tue Sep 24 06:55:15 2002

(c) Copyright 2000 Oracle Corporation. All rights reserved.

DBMS Consulting, Inc. Sunil G. Singh
dbmssunserver3% more ref_codelist_error.rec
SELECT E.MODULE_EXECUTION_ID,E.MODE_OF_EXECUTION_CODE,E.OUTPUT_DEVICE_TYPE_CODE,
E.MODULE_TYPE_CODE,E.CHECK_DB_STATE_FLAG,E.DATABASE_ROLE,E.OUTPUT_WIDTH FROM MODULE_EXECUTIONS E,MODULES M WHERE M.NAME = UPPER(:b1) AND M.MODULE_ID = E.MODULE_ID AND TASK_NAME = :b2;
SELECT NVL(O.DEFAULT_REPORT_RS,RRS.LONG_VALUE),NVL(O.DEFAULT_RS_PRINTER,RPRINTER.LONG_VALUE) FROM REFERENCE_CODELITS_VALUES RRS,REFERENCE_CODELITS_VALUES RPRINTER,ORACLE_ACCOUNTS O WHERE O.ORACLE_ACCOUNT_NAME = USER AND RRS.REF_CODELITS_VALUE = 'OCL_JOB_PREF' AND RRS.REF_CODELITS_VALUE_SHORT_VAL = :b1 AND RPRINTER.REF_CODELITS_VALUE = 'OCL_JOB_PREF' AND RPRINTER.REF_CODELITS_VALUE_SHORT_VAL = :b2;
SELECT USER FROM SYS.DUAL;
SELECT LONG_VALUE FROM REFERENCE_CODELITS_VALUES WHERE REF_CODELITS_NAME = 'OCL_STATE' AND REF_CODELITS_VALUE_SHORT_VAL = 'PRINTER_TYPE' AND ACTIVE_FLAG = 'Y';
SELECT DESCRIPTION,LONG_VALUE FROM REFERENCE_CODELITS_VALUES WHERE REF_CODELITS_NAME = 'REPORT_SERVER' AND REF_CODELITS_VALUE_SHORT_VAL = :b1;
SELECT DESCRIPTION,LONG_VALUE FROM REFERENCE_CODELITS_VALUES WHERE REF_CODELITS_NAME = 'PRINT_QUEUE_NAME' AND REF_CODELITS_VALUE_SHORT_VAL = :b1;
SELECT REPLACING_MESSAGE_TOPIC_ID,POPUP_FLAG,SINGLE_LINE_TEXT,MESSAGE_SEVERITY_TYPE_CODE FROM MESSAGE_TOPICS WHERE :b1 = MESSAGE_TOPIC_ID;
SELECT DESCRIPTION,LONG_VALUE FROM REFERENCE_CODELITS_VALUES WHERE REF_CODELITS_NAME = 'PRINT_QUEUE_NAME' AND REF_CODELITS_VALUE_SHORT_VAL = :b1

DBMS Consulting, Inc. Sunil G. Singh
OCUG San Diego 2002: OPA Debugging Techniques in a 4.0 Environment

dbmssunserver3% more ref_codelist_error.trc

PARSING IN CURSOR #19 len=148 dep=0 uid=136 oct=3 lid=136 tim=0 hv=3554033569 ad = '82e1c4b8'
SELECT DESCRIPTION,LONG_VALUE FROM REFERENCE_CODELIST_VALUES WHERE REF_CODELIST_NAME = 'PRINT QUEUE NAME' AND REF_CODELIST_VALUE_SHORT_VALUE = :b1
END OF STMT

PARSE #19:c=0,e=0,p=0,cr=0,cu=0,mis=1,r=0,dep=0,og=0,tim=0
WAIT #19: nam='SQL*Net message to client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #19: nam='SQL*Net message from client' ela= 0 p1=675562835 p2=1 p3=0
BINDS #19:
  bind 0: dty=1 mxl=128(50) mal=00 scl=00 pre=00 oacflg=03 oacfl2=10 size=128 off set=0
                      bfp=01a40310 bln=128 avl=13 flg=05
                      value="%RXC_PRINTER%"
EXEC #19:c=0,e=0,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=4,tim=0
FETCH #19:c=0,e=0,p=0,cr=2,cu=0,mis=0,r=0,dep=0,og=4,tim=0
WAIT #19: nam='SQL*Net message to client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #19: nam='SQL*Net message from client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #19: nam='SQL*Net message to client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #19: nam='SQL*Net message from client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #21: nam='SQL*Net message to client' ela= 0 p1=675562835 p2=1 p3=0
WAIT #21: nam='SQL*Net message from client' ela= 0 p1=675562835 p2=1 p3=0

DBMS Consulting, Inc. Sunil G. Singh
OPA Forms Level: debug_toggle=Y

- Real-life example: Admin -> Glib Admin -> User Domain. Click [All Domains] button, click exit, answer Yes to question “Would you like to save the changes you have made?”
- Returns error FRM-40105 – Unable to resolve reference to item CONTROL.SAVE.
- It happens that this error occurs without much contact/execution on the RDBMS level
Domain Lists

Domain List Sub Type: User
Name: OPS$OPAPPS

Domain:
- STANDARD
- QA
- QA2

Forms:
FRM-40105 - Unable to resolve reference to item CONTROL.SAVE.
Domain Lists

Domain List Sub Type: User
Name: OPS$OPAPPSS

Domain
- STANDARD
- QA
- QA2

Forms
FRM-40700 - No such trigger: WHEN-BUTTON-PRESSED.
The debug_toggle is an old parameter from the earlier versions of OC. It is also used for debugging ftp file transfer failures.

In the %ORACLE_HOME%\806\forms60\server\formsweb.cfg file, add debug_toggle=Y to the end of the line which starts with form= in the [opa40] section.

Stop and restart the OracleiSuitesHTTPServer and the Oracle Forms Server [Forms60Server] service.

In a production environment, users may have to log off for a short period of time for this step.
In a query mode form, hit SHIFT+F8. There will be a message on the lower left screen status bar that says “Debug Level is now 1”.

Hit SHIFT+F8 again, the Debug Level will increment by 1. The maximum value is 5.

In some Forms, SHIFT+F8 will bring up a dialogue box for printing first. Try a query in Admin -> Reference Codelists -> Local Codelists and then SHIFT_F8. Then set the debug level and go to the menu path in question.
**OS Level: Strings the Forms .fmx and .rep files**

- Sometimes, even the debug messages are not enough to definitively determine where a problem is.
- Action -> Environment in most parts of OC will say what module is running, and this is sometimes the name of the actual form.
- ftp `%OPA_HOME%\oc\<module>.fmx` to a UNIX machine and perform
  - `strings <module.fmx> | grep -i "<Error Message String>`
- Sometimes, finding the source of an error message or a particular select statement can be a problem, and a partial clue as to the logic happening at that particular time of failure is crucial for an analysis.
Sometimes, finding the source of an error message or a particular select statement can be a problem, and a partial clue as to the logic happening at that particular time of failure is crucial for an analysis.

Performing a strings on the binary file running the module and “grep”ing the error or SQL statement can be very useful, especially for:

- `rxcbbeblt` (Batch Loading)
- `rxcbvbs` (Batch Validation)
- `Rxcdxbvb` (Data Extract Views)
**OPA Level: Set .oclrc parameters**

- Most batch jobs will give more verbose output if `$RXC_DEBUG` is set in .oclrc. Some will also provide traces in `USER_DUMP_DEST` if `$SQL_TRACE` is set in .oclrc
  - `RXC_DEBUG=3`; export `RXC_DEBUG`
  - `SQL_TRACE=TRUE`; export `SQL_TRACE`

- Recall that .oclrc is no longer necessary in OC 4.0.x, so this file has to be manually created in the user’s `$HOME` directory. Additionally, this file should obey Bourne shell syntax since `rxcpsdps` runs in `/bin/sh`. 
OS Level: Run the command from Batch_Jobs

- Sometimes, some batch jobs might core and not leave a complete log file. In this case, it is very useful to query the cmd_buffer from the RXC.BATCH_JOBS table and manually execute this command from the UNIX command line logged in as the user in question.
- The parameters for the logfile and outfile will have to be adjusted, usually something in /tmp, to prevent a file conflict with the files already existing from the previous failed run.
- This cmd_buffer was previously in OC 3.x log files, but somehow is no longer present in 4.0.x
OS Level: Trap the at file

- If a particular batch job does not execute correctly and always results in a mail message with some strange failure to the rxcprod account, the at job which runs this job can be trapped.
- Schedule the job in question for sometime in the future.
- Examine /var/spool/cron/atjobs directory and search for a file ending in .a.
- Examine this file to determine if there is a possible shell script syntax error, or possibly an environment variable that is not being set correctly.
**OPA Level: OPA_TRACE.DebugOn**

- The function OPA.OPA_TRACE.DebugOn can be called for some packages (mostly TMS) which are wrapped but sometimes return errors.
- The output is generally very verbose so it is best to have a small set of test data to reproduce the issue.
- Set serveroutput on size 1000000 is a prerequisite, as well as a manual call to the procedure/function in question, where the arguments have to first be determined.
- The TableOn procedure can also be called to populate the table OPA.OPA_DEBUG for tests that are very large.
- This is particularly useful for derivation issues from OC to TMS. A list of packages where OPA_TRACE can be used appears on the next slide.
<table>
<thead>
<tr>
<th>OPA_USER_MENU</th>
<th>OPA_USER_MENU_ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPA_USER_MODS</td>
<td></td>
</tr>
<tr>
<td>DISCREP_MGMT</td>
<td></td>
</tr>
<tr>
<td>OCJLE</td>
<td></td>
</tr>
<tr>
<td>RXC</td>
<td>RXCRRFTRACKDATA</td>
</tr>
<tr>
<td>TMS_OCL_DERV</td>
<td>TMS_ACTIVATION_CONTENTS_M1</td>
</tr>
<tr>
<td>TMS_ACTIVATION_RELATIONS_M1</td>
<td>TMS_CLASSIFICATION_INFO</td>
</tr>
<tr>
<td>TMS_CONFLICT_RES</td>
<td>TMS_HIERARCHY</td>
</tr>
<tr>
<td>TMS_HL_OMISSIONS_M1</td>
<td>TMS_MASTER_ACTION</td>
</tr>
<tr>
<td>TMS_MASTER_CLASSIFICATION</td>
<td>TMS_MASTER_SYNCHRONIZATION</td>
</tr>
<tr>
<td>TMS_PREDICT_INFO_HDRS_M1</td>
<td>TMS_SOURCE_TERMS_M1</td>
</tr>
<tr>
<td>TMS_UD_ACTIVATION_RULES</td>
<td>TMS_USER_ACTIVATION</td>
</tr>
<tr>
<td>TMS_USER_AUTOCODE</td>
<td>TMS_USER_DEF_ACTION</td>
</tr>
<tr>
<td>TMS_USER_DEF_DICTIONARY</td>
<td>TMS_USER_FULLAUTOCODE</td>
</tr>
<tr>
<td>TMS_USER_MT_DICTIONARY</td>
<td>TMS_USER_MT_ERROR_LOG</td>
</tr>
<tr>
<td>TMS_USER_HT_HL_CLASSIFICATIONS</td>
<td>TMS_USER_RECLASSIFICATION</td>
</tr>
<tr>
<td>TMS_USER_RUN</td>
<td>TMS_USER_SEARCHOBJECT</td>
</tr>
<tr>
<td>TMS_USER_SQLVALIDATION</td>
<td>TMS_USER_SYNCHRONIZATION</td>
</tr>
<tr>
<td>TMS_USER_OMISSIONS_M1</td>
<td></td>
</tr>
</tbody>
</table>
Q&A.

- Please come to Booth 19 and 20 with any additional questions, and also for:
  - Additional copies of this Presentation
  - Free CD Cases
  - More Brochures, posters and cards
  - US Flag pin