Oracle 11g: Learning to Love the ADR

Presented by: Don Seiler

Pythian
love your data

About Don Seiler

- From Manitowoc, Wisconsin
- Oracle DBA since 2001
- Started with Oracle 7.3.4 on HP-UX
- With Pythian since May 2008
- Blogger - www.seiler.us

Why Companies Trust Pythian

- Recognized Leader:

- Global industry-leader in remote database administration services and consulting for Oracle, Oracle Applications, MySQL and SQL Server
- Work with over 150 multinational companies such as Forbes.com, Fox Interactive Media, Nordion and Western Union to help manage their complex IT deployments

- Expertise:

- One of the world's largest concentrations of dedicated, full-time DBA expertise. Employ 6 Oracle ACEs/ACE Directors.
- Specialized in Oracle Database, RAC, Data Warehousing, Performance Tuning, Oracle Exadata, GoldenGate and Enterprise Linux.

- Global Reach & Scalability:

- 24/7/365 global remote support for DBA and consulting, systems administration, special projects or emergency response

Diagnostic Files

Pythian
love your data

What Are We Talking About?

- Alert Log (RDBMS, ASM, Listener)
- Trace Files
  - User-Generated (10046, 10053)
  - System-Generated (ORA error)
- Core Dumps

Diagnostic Files in 10g

- $ORACLE_BASE/admin/$ORACLE_SID/
  - bdump – alert log & background trace files
  - cdump – core dump files
  - udump – user trace files (10046, 10053)
  - adump – audit files

These are defaults, overridden with %_dump_dest parameters.

Diagnostic Files in 11g

- Automatic Diagnostic Repostory (ADR)!
- Specified by DIAGNOSTIC_DEST
  - New initialization parameter, replaces %_dump_dest
    - If not set, defaults to ORACLE_BASE
      - If ORACLE_BASE not set, defaults to ORACLE_HOME/log
  - First creates a “diag” directory inside its home.
  - Check V$DIAG_INFO for location details

V$DIAG_INFO

```
SQL> select name, value from v$diag_info;

NAME                      VALUE
------------------------- ------------------------------------------------------------
Diag Enabled              TRUE
ADR Base                  c:\oracle
ADR Home                  c:\oracle\diag\rdbms\orcl\orcl
Diag Trace                c:\oracle\diag\rdbms\orcl\orcl\trace
Diag Alert                c:\oracle\diag\rdbms\orcl\orcl\alert
Diag Incident             c:\oracle\diag\rdbms\orcl\orcl\incident
Diag Cdump                c:\oracle\diag\rdbms\orcl\orcl\cdump
Health Monitor            c:\oracle\diag\rdbms\orcl\orcl\hm
Default Trace File        c:\oracle\diag\rdbms\orcl\orcl\trace\orcl_ora_4084.trc
Active Problem Count      0
Active Incident Count     0

11 rows selected.
```

What's in the ADR?

- Product Type: asm / rdbms / tnslsnr / clients
- Both human-readable (trace) and XML alert logs
- Trace files (10046 etc. in trace directory)
  - Including SQLNet trace files
- Incident files
- Core dump files

Disabling the ADR

- Set hidden parameter:

```
alter system set "_diag_adr_enabled"=false
scope=spfile;
```

- Be sure to set the old *_dump_dest parameters
- Not advised (obviously)!

Too Many Trace Files!

- First thing I noticed after upgrading to 11g!
- Can be greatly reduced with hidden parameter:
  - Set _disable_health_check = TRUE
  - Don’t set in production without consulting Oracle Support!

Quick Terminology Lesson

- Problem – A critical error in the database (e.g. ORA-00600, 07445, 04031)
- Incident – A single occurence of a problem.
- Follows ITIL!

ADR Benefits & ADRCI

Pythian
love your data

ADRCI

```
$ adrci

adrci> show base;

ADR base is "/u01/app/oracle"


adrci> show homes;

ADR Homes:

diag/tnslsnr/myhost/listener

diag/asm/+asm/+ASM

diag/rdbms/mydb/mydb


adrci> set home diag/rdbms/mydb/mydb;
```

Using ADRCI to View Alert Log

- View / Tail alert log (even on Windows!)
  - `adrci> show alert -tail -f;`
  - Read from XML files.
  - Results opened in text editor unless -term specified.
- View alert log contents matching patterns.
  - `adrci> show alert -p "message_text like '%ORA-%'" -term;`
  - Includes message timestamps, better than grep.
  - Case-sensitive!
  - Can pattern-match on any predicate field such as host_id, user_id, originating_timestamp, etc.

Viewing Alert Log by Timestamp

- `adrci> show alert -p "originating_timestamp >= systimestamp-1/24" -term;`
- `adrci> show alert -p "message_text like '%ORA-600%' and originating_timestamp >= systimestamp-30" -term;`
- Obvious alert log monitoring benefits!
- Requires presence of XML log files.
- Can SPOOL output to new log files.

Viewing DDL in Alert Log

- Enable DDL Logging:

```
alter system set enable_ddl_logging=true scope=both;

alter session set enable_ddl_logging=true;
```

- View output:

```
adrci> show alert -p "message_text like '%DROP%'" -term;

ADR Home = c:\oracle\diag\rdbms\orcl\orcl:

*************************************************************************

2011-08-16 15:33:49.805000 -05:00

DROP TABLE DTS_OBJECTS2
```

  - Reminder: Logs and search are case-sensitive!

File Maintenance

- Alert log XML files rotated after 10M.
- Old Alert log XML files and trace/incident files purged depending on short or long purge policies.
- Trace (human-readable) alert and listener log files not rotated or archived by Oracle.

Short Purge Policy

- `adrci> show control;`
- SHORTP_POLICY
- Value in HOURS, default is 720 (30 days)
- Max is 35791394 (over 4000 years!)
- Setting to 0 (zero) means all “short life” files can be purged.
- Trace files, core dumps, packaging info
- `set control (SHORTP_POLICY = 336);`

Long Purge Policy

- LONGP_POLICY
- Default is 8760 (365 days)
- Same max & zero rules as SHORTP_POLICY
- Incident info, incident dumps, alert logs
- `set control (LONGP_POLICY = 1440);`

Manually Purging Files

- Purge alert logs older than 1 day
  - -age parameters takes minutes

```
adrci> purge -age 1440 -type alert;
```

- Purge all files associated for an incident

```
adrci> purge -i 12345;
```

- For more

```
adrci> help purge;
```

ADR Purging Bug!

- **Bug 9500575**: 11G LISTENER LOGS LOG_XYZ.XML ARE NOT AUTOMATICALLY PURGED
  - Potential to fill up diagnostic_dest disk.
  - Oracle 11.1.0.7 and 11.2.0.1
  - To manually purge files older than 1 week:
    - `adrci> purge -age 10080 -type alert;`

Incident Packaging Service

Pythian
love your data

Incident Packaging Service (IPS)

- Creates logical “packages” for Oracle Support based on criteria such as incident number or timeframe.
- Either via EM or adrci

```
adrci> show incident;

adrci> show problem;

adrci> show tracefile [-i 123] [-[r]t];
```

Incident Flood Control

- Flood-controlled incidents will generate an alert log entry but not incident dumps.
- Thresholds:
  - 5 incidents for the same problem key in one hour. Will resume in the next hour.
  - 25 incidents for the same problem key in one day. Will resume in the next day.
  - 50/hour or 250/day = No more incident recording at all until hour/day expires!

IPS – Creating Packages

```
adrci>IPS CREATE PACKAGE INCIDENT inc_number;

adrci>IPS CREATE PACKAGE PROBLEM problem_ID;

adrci>IPS CREATE PACKAGE PROBLEMKEY "problem_key";

adrci>IPS CREATE PACKAGE SECONDS sec;

adrci>IPS CREATE PACKAGE TIME 'start_time' TO 'end_time';

adrci>IPS CREATE PACKAGE;
```

- Use `IPS ADD INCIDENT` or `IPS ADD FILE` to add data to the package before generating it.

IPS – Generating Package Files

- Creates a physical file from the logical package.

```
adrci>IPS GENERATE PACKAGE package_number IN path

adrci>IPS GENERATE PACKAGE 1 IN /home/seiler/diagnostics

adrci>IPS SHOW FILES PACKAGE 1
```

- Create and generate package with one command: IPS PACK

```
adrci>IPS PACK INCIDENT incident_id IN path;

adrci>IPS PACK INCIDENT 12345 IN /home/seiler/diagnostics;
```

ADRCI Scripts

Pythian
love your data

ADRCI Scripts

- Can be called interactively a la sqlplus:
  - `adrci> @/home/seiler/scripts/test.adrci`
- Can be called on command-line:
  - `$ adrci script=/home/seiler/scripts/test.adrci`
- Example script:

```
# ADRCI script to find alert log errors

SPOOL /home/seiler/logs/alert_log_errors.log

ECHO "ALERT LOG ERRORS:"; SET HOMEPATH diag/rdbms/orcl/orcl; SHOW
ALERT -TERM -P "MESSAGE_TEXT LIKE '%ORA-%'"

SPOOL OFF
```

ADRCI Reports

Pythian
love your data

ADRCI Health Reports

- Only applies to Health Monitor for now

```
adrci> show hm_run; # Look for RUN_NAME, e.g. HM_RUN_1

adrci> create report hm_run hm_run_1;

adrci> show report hm_run hm_run_1; # Outputs XML
```

- Human-readable report:

```
SQL> set long 99999 longchunksize 1000 pagesize 999 linesize 512

SQL> select dbms_hm.get_run_report('HM_RUN_1') from dual;
```

Example Incident Scenario

Pythian
love your data

Step 1: Cause an Incident!

```
SQL> alter session set events '942 incident(table_missing)';

Session altered.


SQL> drop table doesnotexist;

drop table doesnotexist
           *

ERROR at line 1:

ORA-00942: table or view does not exist


SQL> alter session set events '942 trace name context off';

Session altered.
```

Alert Log of Incident

```
Fri Aug 12 15:44:54 2011

Errors in file
/home/oracle/app/oracle/diag/rdbms/orcl/orcl/trace/orcl_ora_18457
.trc  (incident=12201):

ORA-00700: soft internal error, arguments:
[EVENT_CREATED_INCIDENT], [942], [TABLE_MISSING], [], [], [], [],
[], [], [], [], []

ORA-00942: table or view does not exist

Incident details in:
/home/oracle/app/oracle/diag/rdbms/orcl/orcl/incident/incdir_1220
1/orcl_ora_18457_i12201.trc
```

Alert Log via ADRCI

```
adrci> show alert -p "message_text like '%ORA-00942%'" -term

adrci> show alert -p "message_text like '%ORA-%' and
originating_timestamp >= systimestamp-1/24" -term

ADR Home = c:\oracle\diag\rdbms\orcl\orcl:

*************************************************************

2011-08-14 14:35:27.843000 -05:00

Errors in file
c:\oracle\diag\rdbms\orcl\orcl\trace\orcl_ora_4688.trc
(incident=14633):

ORA-00700: soft internal error, arguments:
[EVENT_CREATED_INCIDENT], [942], [TABLE_MISSING], [], [], [], [],
[], [], [], []

ORA-00942: table or view does not exist
```

Step 2: View the Incident

```
adrci> set home diag/rdbms/orcl/orcl

adrci> show incident

ADR Home = /home/oracle/app/oracle/diag/rdbms/orcl/orcl:

*************************************************************************
********

INCIDENT_ID          PROBLEM_KEY
                CREATE_TIME

--------------------
-------------------------------------------------------------
---------------------------------------

12201                ORA 700 [EVENT_CREATED_INCIDENT] [942]
[TABLE_MISSING]      2011-08-12 15:44:54.472000 -05:00

1 rows fetch
```

Step 3: Create a Package

```
adrci> ips create package incident 12201;

Created package 1 based on incident id 12201, correlation level
typical

adrci> ips generate package 1 in /home/oracle

Generated package 1 in file
/home/oracle/ORA700EVE_20110812154951_COM_1.zip, mode complete
```

OR:

```
adrci> ips pack incident 12201 in /home/oracle

Generated package 2 in file
/home/oracle/ORA700EVE_20110812160901_COM_1.zip, mode complete
```

View Package Contents

- Obvious:

```
unzip -l /home/oracle/ORA700EVE_20110812154951_COM_1.zip
```

- The ADRCI Way:

```
adrci> ips show incidents package 1

adrci> ips show files package 1

   FILE_ID                 1
   FILE_LOCATION           <ADR_HOME>/incident/incdir_12201
   FILE_NAME               orcl_ora_18457_i12201.trc
   LAST_SEQUENCE           1
   EXCLUDE                 Included
```

Step 4: Send Package to OSS

- Package will contain:
  - Alert log
  - Incident trace files
  - Export/dmp files
- Hopefully will reduce the file requests from Oracle Support!

Live Demo!

Pythian
love your data

Questions?

Pythian
love your data

Thank You!

- Email: seiler@pythian.com
  - Please write with any questions or criticisms!
- Blogs:
  - http://www.seiler.us
  - http://www.pythian.com
- Twitter: @dtseiler / @pythian
- Follow Pythian on Facebook & LinkedIn!