

## Practice

# Using ADR Command Interpreter

## Practice Target

In this practice, you will execute the common tasks included in using the ADR command interpreter.

## Practice Overview

In this practice, you will perform the following tasks:

- Examine the ADR directory structure for an Oracle database home
- Start `adrci` utility and use its help system
- View the alertlog file from within `adrci`
- Create problem package from `adrci`

## Assumptions

- This practice assumes that `srv1` is up and running from the **CDB** snapshot.

## Caution

- Because you are going to perform a destructive action in this practice on the database, unless the vm is already recently restored from its CDB snapshot, I recommend creating a snapshot for the vm and delete the current snapshot. In the end of the practice, you will restore the vm back from its created snapshot.

## A. Examining ADR Directory Structure

In this section of the practice, you will examine the ADR directory structure, especially for Oracle database home.

1. Open Putty and connect to `srv1` as `oracle`.

2. Invoke SQL\*Plus and login to the root container as `SYS`

```
sqlplus / as sysdba
```

3. Display the value of the parameter `DIAGNOSTIC_DEST`

By default, this parameter is set to the `$ORACLE_BASE`

```
show parameter DIAGNOSTIC_DEST
```

4. Exit from SQL\*Plus and change the current directory to the directory pointed by the parameter `DIAGNOSTIC_DEST`. Identify the directory pointed by the ADR.

ADR is located under the `diag` directory.

```
cd /u01/app/oracle  
ls -al
```

5. Change the current directory to the `diag` directory and list the subdirectories under it. Identify the directory that is linked to Oracle database home.

The directories are denoted to Oracle components and products. The directory that is linked to the Oracle database must be the `rdbms`.

```
cd diag  
ls -al
```

6. Change the current directory to `rdbms` and then change the current directory to the two sub-directories under it. Observe their names.

The two subdirectory names are `oradb`. These are the database name and the database instance name.

```
cd rdbms  
cd oradb  
cd oradb
```

7. Examine the directory structure in the current directory.

Those are the ADR repository directories for Oracle database product.

"alert" subdirectory contains the log.xml file, which is the xml version of the alertlog file.

"trace" subdirectory contains the trace files and the text version of the alertlog file.

```
ls -al
```

8. Change the current directory to the `trace` subdirectory. Look for the alertlog file in the subdirectory.

```
cd trace
ls alert_oradb.log
```

9. Open the alertlog file with a `vi` editor and examine its contents. Close `vi` afterwards.

The log file entries are inserted by chronological order.

Normally, we look for "ORA-" message to check on errors or issues.

```
vi alert_oradb.log
```

10. Examine the other files in the `trace` directory.

You should see files of the extensions `.trc` and `.trm`. From the trace file name, we can know which process generated the trace file. Feel free to view the contents of some trace files.

11. Retrieve the total size of trace directory.

In Linux, `du` command displays the sizes for the current working directory and all sub-directories recursively.

The database keeps generating trace files as long as it is working. If we do not manage the sizes taken by the trace files in the ADR, we will end up with large disk space consumed by enormous trace files. We can manage the size taken by the trace files using the command `purge` in `adrci`.

```
du -h
```

## B. Starting `adrci` and Using its Help System

In this section of the practice, you will start the utility `adrci` and use its help system.

12. Type the following command.

You will see the parameters that we can pass to the `adrci` command.

```
adrci -help
```

13. Start the `adrci` utility to enter into its command prompt.

When `adrci` is started, it displays the ADR base directory. The utility is able to work on all the Oracle homes under that root base directory.

```
adrci
```

14. Issue the following command.

`help` command displays a list of the commands that we can submit in `adrci` command prompt

```
help
```

15. Issue the following command.

`help` command can be used to obtain details on any `adrci` command.

```
help purge
```

16. Display the homes under the base directory.

The command shows the ADR homes in the current ADRCI session. To display all the available homes under the base, pass the option `-ALL` to the command.

```
show homes
```

17. Display the current `homepath`.

There could be more than one current `homepath`.

```
show homepath
```

18. Set the current `homepath` to the Oracle database home.

Observe that there is no equal sign for setting `homepath`.

```
set homepath diag/rdbms/oradb/oradb
```

## C. Viewing alertlog File in adrci

In this section of the practice, you will view the database alertlog file from within the `adrci`.

19. Type the following command.

Observe that you can use a long list of predicate strings on the `show alert` command.

```
help show alert
```

20. Issue the following command.

The alertlog file is opened in the `vi` editor (the default editor).

```
show alert
```

21. Display the last two messages in the alertlog file.

```
show alert -tail 2
```

22. Check if there is any Oracle database server error (ORA-xxx) in the alertlog file.

The command retrieves the messages containing the phrase "ORA-" into a temporary file and view it in `vi` editor.

```
show alert -p "MESSAGE_TEXT LIKE '%ORA-%'"
```

23. Issue the following command to display list of the available predicate strings that can be used in the 'show alert -p' command.

```
help show alert
```



## D. Creating a Problem Package from adrci

In this section of the practice, you will purposely raise an incident in the database. Then you will implement the procedure to create a problem package.

24. Exit from `adrci` command prompt.

```
quit
```

25. Issue the following commands to create a testing tablespace and a testing table.

```
sqlplus system/ABcd##1234@pdb1
CREATE TABLESPACE TEST DATAFILE '/home/oracle/test.dbf' SIZE 10M AUTOEXTEND OFF;
CREATE TABLE TEST
AS SELECT DBMS_RANDOM.STRING('A',10) AS RANDOM FROM DUAL CONNECT BY LEVEL <= 50000;
ALTER SYSTEM FLUSH BUFFER_CACHE ;
```

26. Obtain the block header number of `SYSTEM.TEST` table. Add a small number (any number between 3 and 9) to the obtained number and take a note of the result.

The query returns the block header. You will later corrupt a block that is a few blocks away from this block. For example, if the query returns the number 100, you may corrupt the block number 105.

```
SELECT HEADER_BLOCK FROM DBA_SEGMENTS WHERE OWNER='SYSTEM' AND SEGMENT_NAME='TEST';
```

27. Exit from SQL\*Plus

```
exit
```

28. Use the `dd` utility to make a corruption in the named block.

Substitute the `<datafile full name>` with the datafile full name obtained above.

Substitute the `<block# to corrupt>` with the block number that you noted earlier.

When you execute the command, it will prompt you to enter data. Type any word, like "testme" for example, then press [Enter]. After that, type `EOF` then press [Enter].

```
dd of=/home/oracle/test.dbf bs=8192 conv=notrunc seek=<block# to corrupt> << EOF
```

29. Validate the datafile.

```
rman target sys/ABcd##1234@pdb1
VALIDATE TABLESPACE 'TEST';
```

30. Login to `PDB1` as `SYSTEM` and select all the data from the `TEST` table.

Unfortunately, the incident is not reported by the `adrci` until we try to access the corrupted blocks.

```
sqlplus system/ABcd##1234@pdb1
SELECT * FROM TEST;
exit
```

31. Open the `adrci`

```
adrci
```

32. Set the `homepath` to the Oracle database then display the incidents.

The command should retrieve an incident or more in the database. The command displays the `INCIDENT_ID` and the `PROBLEM_KEY`.

```
adrci> set homepath diag/rdbms/oradb/oradb
show incident
```

33. Display the incidents in `BREIF` mode.

```
show incident -mode brief
```

34. Display the incidents in `DETAIL` mode.

```
show incident -mode detail
```

35. Display the tail of the alertlog file to verify that the incident is reported in it.

```
show alert -tail
```

36. Display the incidents again (to display the `INCIDENT_ID`) then retrieve any trace file associated with it.

```
show incident
show tracefile -i <incident-id>
```

37. Create a package for the reported problem.

Retrieve the problem key from the output of the `show incident` command.

This command creates a logical package and add the required data into its metadata. No file is added yet into the package.

```
ips create package problemkey '<problem-key>'
```

38. Generate the physical incident package into the staging folder.

The package number should be retrieved by the command in the preceding step.

The generated file name is of the format `ORA1578_*.zip`

```
ips generate package 1 IN '/media/sf_staging/'
```

39. Open the package file and explore its contents.

You should see a copy of the database ADR directory structure with the needed files in it.

## Cleanup

40. Shutdown `srv1` and restore it from its **CDB** snapshot.
41. Delete the package file from the sharing folder.





## Summary

- Oracle saves the diagnostic files in a centralized repository called ADR. The files are organized in specific ADR directory hierarchy.
- The `adrci` utility makes it easier to view the alertlog file contents than using the normal editor.
- The `adrci` utility provides easy to use commands to gather the diagnostic files required for troubleshooting incidents that happened in Oracle database. However, not all the database issues are considered as incidents.

