Author – Anand Mandilwar
https://anandmandilwar.com

Re-org of EBS Database using export/Import of complete database
Database Version – 11.2.03 (Hosted on Oracle Linux)
EBS Version – 12.2.2

MOS Document to be followed - Doc ID 741818.1

- **Export the source database instance**
  - Shut down the application services
  - Create a working directory on the DB server node
    - $ mkdir –p /dummy/oracle/expimp
  - Copy $AU_TOP/patch/115/import/auexpdp.dat from the source administration server node to the working directory in the source database server node. Use a text editor to modify the file to reflect the source environment and other customized parameters
  - Parameter modified as per my environment
    - parallel=12
    - filesize=15728640000 (15 G)
  - create a directory called “dmpdir” as defined in export parameter file
    - Connect as sysdba
    - create directory dmpdir as '/dummy/oracle/expimp'
  - Source the DB home and Oracle SID
  - Login to DB server and start the export
    - $ expdp "'/ as sysdba'" parfile=/dummy/oracle/expimp/auexpdp.dat
    - **Issue encountered**
      - ORA-39038: Object path "TABLE" is not supported for SCHEMA jobs OR ORA-39038: Object path "SCHEMA" is not supported for FULL jobs
    - **Solution**
      - Connect as sysdba
      - Run $ORACLE_HOME/rdbms/admin/catmet2.sql
      - Verify if below SQL returns any record
        - SELECT named, object_path, comments FROM schema_export_objects WHERE named='Y';
        - If no, retry export
    - Export finishes successfully (**took around 12 hrs**) – with some exceptions viz (SR 3-13518999651 is already logged with Oracle but no fruitful response so far)
      - ORA-04063: table "APPS_NE.schedulerRule1602_TAB" has errors
      - ORA-04063: table "CSR.CSR_RULES_B" has errors
      - ORA-04063: table "APPS_NE.rcRouteFuncDelay11572_TAB" has errors
      - ORA-04063: table "AZ.AZ_DIFF_RESULTS" has errors

Preparation for target database

- Generate target database instance creation script aucrdb.sql

- Copy below SQLs from $AU_TOP/patch/115/sql/ and put them on the working directory (created above) on the DB server
  - auclondb.sql
  - auque1.sql
  - audb1120.sql
  - ausy1120.sql
  - aujv1120.sql
  - aumsc1120.sql
- Login to DB server, source the ORACLE HOME, set ORACLE SID and login to SQLplus as sysdba.
- Run the script using 'system' to generate → Parameter is 11 b'cos the underlying DB is 11g
  - sqlplus system/[system password] @auclondb.sql **11**
  - It creates **aucrdb.sql** in the same working directory

- Record Advanced Queue settings
  - Run the script using 'system' to generate
    - sqlplus system/[system password] @ auque1.sql.
    - It generates auque2.sql in the same working directory
- Prepare a target database instance
  - Create pfile from spfile – if the existing DB is using spfile
  - Copy the pfile thus generated to the working directory
  - Create another folder to keep all the new .dbf files, control files under the same 'oradata' FS
    - For this purpose, I created one folder called 'oradata1' within '/dummy/oradata'
      - $ cd /dummy/oradata
      - $ **mkdir –p oradata1**
  - Rename the pfile to ensure it matches the SID. For my case - initDUMMY1.ora. "DUMMY1" is the new DB name
  - Modify the pfile to replace all the occurrences of 'DUMMY' to 'DUMMY1' and 'dummy' to 'dummy1' especially below 2-3 parameters
    - db_name
    - log_archive_dest → create another folder for the new DB
    - service_names
  - Create the target database instance (This is done using the same ORACLE_HOME)
    - Modify the "aucrdb.sql" created above to
      - Rename the datafiles location as per your configuration. e.g. in my case I replaced all the occurrences of "/dummy/oradata" with "/dummy/oradata/**oradata1**/"
      - Updated each datafile with the storage parameter – "next extent", "autotensible" & "maxsize"
      - Allocated adequate no. of datafiles depending on the used space in each of the tablespaces (I had the data from the existing database)
      - Ensured the syntax is correct for each of the commands
      - Added 3-4 more logfile groups (this is needed for import)
    - Set ORACLE_HOME → $ export ORACLE_HOME=/dummy/oracle/product/11.2.0.3

- Set ORACLE_SID to new SID (as updated in the pfile). For my case, it was "DUMMY1" → $ export ORACLE_SID=DUMMY1
- Login as sysdba
    - $ sqlplus /nolog
    - $ conn /as sysdba
    - $ startup mount pfile='/dummy/oracle/expimp/initDUMMY1.ora'
        - **Got an error** while starting DB
            - ORA-27102: out of memory
            - Linux-x86_64 Error: 28: No space left on device
        - **Solution**
            - Shut down the existing DB and listener – DUMMY and then start the new DB. This resolved the error.
    - $ @aurcrdb.sql
    - Ensure none of the scripts in the "aucrdb.sql" errored out and all the specified Tablespaces are created properly. In my case "APPS_TS_MEDIA" tablespace was not created as the command was having syntax error.
    - Exit of the SQL
    - Set up the sys schema
        - Run "audb1120.sql" using sysdba
        - sqlplus "/ as sysdba" @audb1120.sql
    - set up SYSTEM schema
        - sqlplus system/[system password] @ausy1120.sql → SYSTEM password is default "manager"

```
SQL> !ls -ltr ausy1120.sql
-rwxr-xr-x 1 ortalqdi dbtalqdi 5252 Oct 25 09:45 ausy1120.sql

SQL> @ausy1120.sql

SQL> @ausy1120.sql

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.



------------------------------------------------------------
--- ausy1120 started at 26-OCT-2016 12:42:53 ---

'---AUSY1120COMPLETEDAT'||TO_CHAR(SYSDATE,'DD-MON-YYYYHH24:M
------------------------------------------------------------
--- ausy1120 completed at 26-OCT-2016 12:42:53 ----


Commit complete.

Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
-bash-3.2$
```

    - InstalL JVM
        - sqlplus system/[system password] @aujv1120.sql
    - Install other required components
        - sqlplus system/[system password] @aumsc1120.sql FALSE SYSAUX TEMP
    - Run below SQL to make sure dba_registry does not have any INVALID component which is valid in other instances.
        - Source ORACLE_HOME

- Set ORACLE_SID=DUMMY1
- Login as sysdba and run – select * from dba_registry where status =’INVALID’
  - Only COMP_ID that was INVALID in new instance but VALID in all other existing environments was – CONTEXT (Oracle Text)
- Followed note - 1906873.1 to make COMP_ID as valid

```
SQL> select name from v$database;

NAME
---------
[redacted]

SQL> !ls -ltr $ORACLE_HOME/ctx/admin/driacc.plb
-rw-rw-rw- 1 [redacted] 5040 Apr 10  2014 /[redacted]/oracle/product/11.2.0.3/ctx/admin/driacc.plb

SQL> @$ORACLE_HOME/ctx/admin/driacc.plb

Package body created.

SQL> set serveroutput on
SQL> execute sys.validate_context;

PL/SQL procedure successfully completed.

SQL> select comp_name, status, substr(version,1,10) as version from dba_registry where comp_id = 'CONTEXT';

COMP_NAME
--------------------------------------------------------------------------------
STATUS
--------------------------------------------
VERSION
----------
Oracle Text
VALID
11.2.0.3.0

COMP_NAME
--------------------------------------------------------------------------------
STATUS
--------------------------------------------
VERSION
----------
```

  - Conn as sysdba
    - $ ALTER SESSION SET CURRENT_SCHEMA=SYS;
    - $ grant select on SYS.DBA_PROCEDURES to ctxsys;
    - $ ALTER SESSION SET CURRENT_SCHEMA=CTXSYS;
    - $ SELECT 'Calling ctx/admin/driacc.plb on ' || SYSTIMESTAMP FROM dual;
    - $ @@?/ctx/admin/driacc.plb;
    - $ execute sys.validate_context;
    - $ select comp_name, status, substr(version,1,10) as version from dba_registry where comp_id = 'CONTEXT'; → it is VALID now. **I was good to run the import now**

- **Import the application database instance**
  - Increase the parallel max server parameter to 20 for the new DB –DUMMY1
  - Login to DUMMY1 as sysdba and run
    - alter system set **parallel_max_servers**=20 scope=memory; → ‘memory’ because we are using pfile to start the DB
  - Provision for additional logfile group – this was taken care in the ‘aucrdb.sql’ itself. I added 3 more groups with 2 members in each group
  - Copy the $AU_TOP/patch/115/import/auimpdp.dat from application server to the working directory on DB server
  - Modify the parameter file to include below parameters

- parallel=12
- EXCLUDE=INDEX,STATISTICS,CONSTRAINTS
- dumpfile=aexp%U.dmp

- Create a directory in the system schema with the name set to the directory specified in the template and the path set to where the export dump files will reside
  - Login to new DB –DUMMY1
  - sqlplus system/[system password]
    - create directory dmpdir as '/dummy/oracle/expimp'
- Source ORACLE_HOME, set ORACLE_SID=DUMMY1 and kick off IMPORT
  - $ impdp "'/ as sysdba'" parfile=/dummy/oracle/expimp/auimpdp.dat → **this runs for 8-10 hrs**
  - 
```
SQL> conn system
Enter password:
Connected.
SQL> create directory dmpdir as '[redacted]/oracle/expimp';

Directory created.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
-bash-3.2$
-bash-3.2$ echo $ORACLE_HOME
[redacted]/oracle/product/11.2.0.3
-bash-3.2$ echo $ORACLE_SID
[redacted]
-bash-3.2$
-bash-3.2$ impdp "'/ as sysdba'" parfile=[redacted]oracle/expimp/auimpdp.dat

Import: Release 11.2.0.3.0 - Production on Wed Oct 26 16:34:01 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
```

- **Import INDEXES**
  - Create a new parameter file "**ind_sqlfile.par**" with the following parameters.
    - INCLUDE=INDEX
    - SQLFILE=createindx.sql
  - Source the ORACLE_HOME
  - Set ORACLE_SID to DUMMY1
  - Run the import using below command. This will generate a file called "createindx.sql" in the same working directory (/dummy/oracle/expimp)
    - impdp "'/ as sysdba'" parfile=/dummy/oracle/expimp/ind_sqlfile.par
    - 
```
-bash-3.2$
-bash-3.2$ ls -ltr [redacted]/oracle/expimp/ind_sqlfile.par
-rw-r--r-- 1 ortalqdi dbtalqdi 102 Oct 27 09:07 [redacted]/oracle/expimp/ind_sqlfile.par
-bash-3.2$
-bash-3.2$
-bash-3.2$ impdp "'/ as sysdba'" parfile=[redacted]/oracle/expimp/ind_sqlfile.par

Import: Release 11.2.0.3.0 - Production on Thu Oct 27 09:12:32 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserv

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Master table "SYS"."SYS_SQL_FILE_FULL_01" successfully loaded/unloaded
Starting "SYS"."SYS_SQL_FILE_FULL_01":  "/******** AS SYSDBA" parfile=[redacted]/oracle/expimp/ind_sqlfile.par
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/INDEX
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/FUNCTIONAL_INDEX/INDEX
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/FUNCTIONAL_INDEX/INDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/BITMAP_INDEX/INDEX
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/STATISTICS/BITMAP_INDEX/INDEX_STATISTICS
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/INDEX/DOMAIN_INDEX/INDEX
Job "SYS"."SYS_SQL_FILE_FULL_01" successfully completed at 09:18:09

-bash-3.2$
```
  - Open the "createindx.sql" file and replace every occurrences of "PARALLEL 1" with "PARALLEL 20" → This will speed up the INDEX creation.
  - Login to new DB – DUMMY1 as sysdba and run the script - createindx.sql
    - $ sqlplus "/as sysdba"
    - $ @ createindx.sql → **This runs for 4-5 hrs**

```
SQL>
SQL> set echo on
SQL> set timing on
SQL> !pwd
/oracle/expimp

SQL> spool createindx.log
SQL> @createindx.sql
```

```
VNC Viewer
 95  ctxsys.driimp.set_value('STOP_WORD','THEY');
 96  ctxsys.driimp.set_value('STOP_WORD','THIS');
 97  ctxsys.driimp.set_value('STOP_WORD','TO');
 98  ctxsys.driimp.set_value('STOP_WORD','UP');
 99  ctxsys.driimp.set_value('STOP_WORD','WAS');
100  ctxsys.driimp.set_value('STOP_WORD','WE');
101  ctxsys.driimp.set_value('STOP_WORD','WERE');
102  ctxsys.driimp.set_value('STOP_WORD','WHEN');
103  ctxsys.driimp.set_value('STOP_WORD','WHICH');
104  ctxsys.driimp.set_value('STOP_WORD','WHO');
105  ctxsys.driimp.set_value('STOP_WORD','WILL');
106  ctxsys.driimp.set_value('STOP_WORD','WITH');
107  ctxsys.driimp.set_value('STOP_WORD','WOULD');
108
109  ctxsys.driimp.set_object('STORAGE','BASIC_STORAGE',2);
110  ctxsys.driimp.set_value('R_TABLE_CLAUSE','lob (data) store as (cache)');
111  ctxsys.driimp.set_value('I_INDEX_CLAUSE','compress 2');
112
113  commit;
114  COMMIT;
115  END;
116  /

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.02
SQL> CREATE INDEX "JTF"."JTF_AMV_ITEMS_URL_CTX" ON "JTF"."JTF_AMV_ITEMS_B" ("URL_STRING")
  2    INDEXTYPE IS "CTXSYS"."CONTEXT"  PARAMETERS ('datastore apps.jtf_url_datastore filter  ctxsys.null_filter section group ctxsys.html_section_group
exer ctxsys.default_lexer language column language_code stoplist ctxsys.default_stoplist storage ctxsys.default_storage wordlist ctxsys.default_wordlist'
PARALLEL 20 ;

Index created.

Elapsed: 00:00:00.41
SQL>
SQL>  ALTER INDEX "JTF"."JTF_AMV_ITEMS_URL_CTX" NOPARALLEL;

Index altered.

Elapsed: 00:00:00.01
SQL> spool off
SQL>
```

- **Import CONSTRAINTS**
  - Create a new parameter file "**constraint_sqlfile.par**" with the following parameters.
    - INCLUDE=CONSTRAINT
    - SQLFILE=createconstr.sql
    - directory=dmpdir
    - dumpfile=aexp%U.dmp
    - logfile=create_const_ddl.log
  - Source the ORACLE_HOME
  - Set ORACLE_SID to DUMMY1
  - Run the import using below command. This will generate a file called "createconstr.sql"
  - Login to new DB – DUMMY1 as sysdba and run the script - createindx.sql
    - $ sqlplus "/as sysdba"
    - $ @ createconstr.sql→ **This runs for 30 min**

```
bash-3.2$
bash-3.2$ impdp "'/ as sysdba'" parfile=/[redacted]/oracle/expimp/constraint_sqlfile.par

Import: Release 11.2.0.3.0 - Production on Thu Oct 27 14:19:42 2016

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
Master table "SYS"."SYS_SQL_FILE_FULL_01" successfully loaded/unloaded
Starting "SYS"."SYS_SQL_FILE_FULL_01":  "/******** AS SYSDBA" parfile=[redacted]/oracle/expimp/constraint_sqlfile.par
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/CONSTRAINT
Processing object type DATABASE_EXPORT/SCHEMA/TABLE/CONSTRAINT/REF_CONSTRAINT
Job "SYS"."SYS_SQL_FILE_FULL_01" successfully completed at 14:20:48

bash-3.2$
```

```
bash-3.2$
bash-3.2$ sqlplus "/as sysdba"

SQL*Plus: Release 11.2.0.3.0 Production on Thu Oct 27 14:26:24 2016

Copyright (c) 1982, 2011, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> spool create_constraint.log
SQL> set echo on
SQL> set timing on
SQL> select name,user from v$database;

NAME      USER
--------- ------------------------------
[redacted]    SYS

Elapsed: 00:00:00.01
SQL> !pwd
/[redacted]oracle/expimp

SQL> @createconstr.sql
```

- Reset Advanced Queues
  - Login as sysdba and run auque2.sql
- Run adgrants.sql
  - Copy $APPL_TOP/admin/adgrants.sql from application server to DB server in the working directory
  - Login as sysdba and run the script - adgrants.sql APPS

```
Creating PL/SQL Package AD_DBMS_METADATA.

Package created.

Package body created.

End of Creating PL/SQL Package AD_DBMS_METADATA.

Creating PL/SQL Package AD_ZD_SYS.

Package created.

Package body created.

End of Creating PL/SQL Package AD_ZD_SYS.
Start of giving grants. This may take few minutes.
PL/SQL procedure successfully completed.

Start of PURGE DBA_RECYCLEBIN.

PL/SQL procedure successfully completed.

End of PURGE DBA_RECYCLEBIN.

Commit complete.

Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
bash-3.2$
```

- Grant create procedure privilege on CTXSYS
  - Copy $AD_TOP/patch/115/sql/adctxprv.sql from the administration server node to the database server node
  - Connect to new DB – DUMMY1 as APPS user (password will be same as that of the existing APPS user in old DUMMY) and run the script
    - $ sqlplus apps@dummy1
    - $ @adctxprv.sql [SYSTEM password] CTXSYS → SYSTEM password will be the same as that of DUMMY
- **Change the DBID and DBNAME**
  - Shutdown the database
    - Source ORACLE_HOME and set ORACLE_SID=DUMMY1
    - $sqlplus "/as sysdba"
    - Shutdown immediate
  - Start the DB in mount mode
    - startup MOUNT PFILE='/dummy/oracle/expimp/initDUMMY1.ora'
    - exit out of SQL
  - Run "nid" – ensuring ORACLE_HOME is correct and ORACLE_SID=DUMMY
    - $ nid TARGET=sys DBNAME=DUMMY LOGFILE=change_dbname.log → it will prompt for password. Provide 'system' password of the existing DUMMY DB.
    - It will take 2-3 min and the command will finish
    - Look into the log file. You will find entries like

- *Database name changed to DUMMY*
- *Database ID for database DUMMY changed to 2700263157*
- *Succesfully changed database name and ID*
- *DBNEWID - Completed successfully*

- Shut down the database
- Rename the pfile (/dummy/oracle/expimp/initDUMMY1.ora) to match it with the new DB name – DUMMY. Hence new parameter file is - /dummy/oracle/expimp/initDUMMY.ora
- Replace all the occurrences of 'DUMMY1' with 'DUMMY' in the parameter file

- Start the new renamed database instance – DUMMY
  - Source the ORACLE_HOME
  - Set ORACLE_SID=DUMMY
  - $ sqlplus /nolog
  - $SQL> conn /as sysdba
  - $SQL> Startup MOUNT pfile='/dummy/oracle/expimp/initDUMMY.ora'
  - $SQL>alter database open resetlogs
  - $ SQL> select dbid, name from v$database;

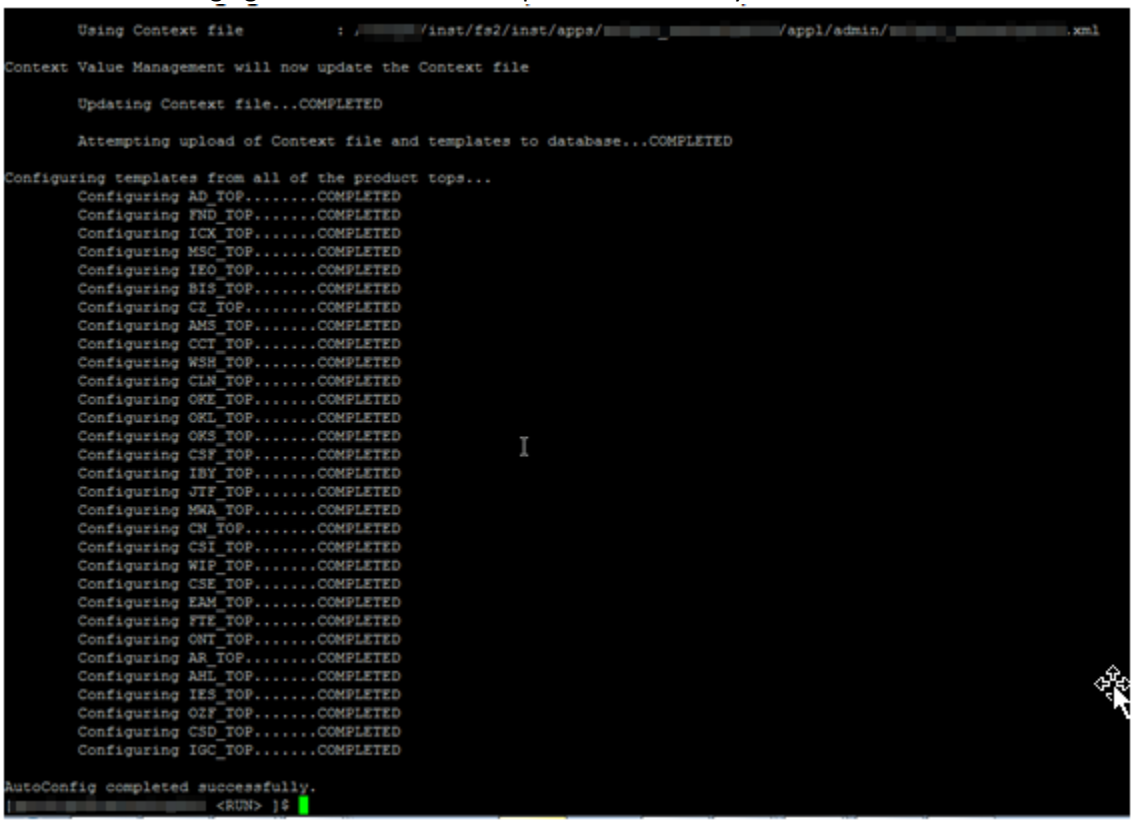
- **Run autoconfig on DB and APPS tier**
  - **DB Tier**
    - cd to $ORACLE_HOME/appsutil/scripts/$CONTEXT_NAME
    - run the script - adautocfg.sh
      - $cd $ORACLE_HOME/appsutil/scripts/$CONTEXT_NAME
      - $ ./ adautocfg.sh → in my case this completed successfully.
  - APPS Tier
    - $ cd $ADMIN_SCRIPTS_HOME
    - $ ./ adautocfg.sh → This errored out and on reviewing the autoconfig log, it showed an error
      - ***AutoConfig could not successfully execute the following scripts:***
      - ***Directory: /dummy/inst/fs2/inst/apps/dummy_host_name/admin/install jtfictx.sh INSTE8_PRF***
    - **Troubleshooting**
      - $ cd /dummy/inst/fs2/inst/apps/dummy_host_name/admin/install
      - $./jtfictx.sh

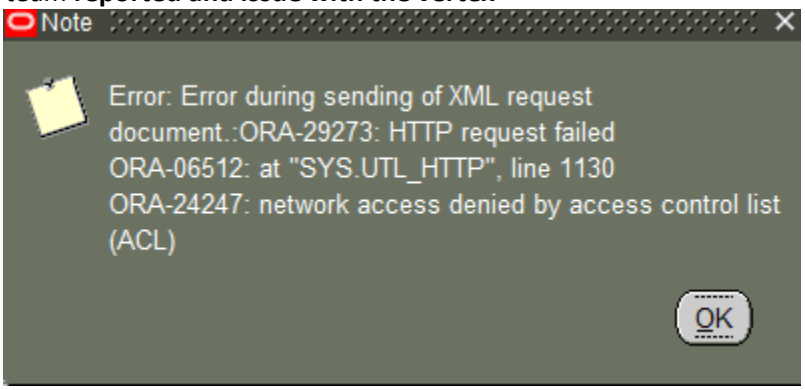
```
[aptalqdi@vmohsalqd020 <RUN> ]$ ./jtfictx.sh

jtfictx.sh started at Fri Oct 28 12:48:06 CDT 2016

APPS username: apps

APPS password:

SQL*Plus: Release 10.1.0.5.0 - Production on Fri Oct 28 12:48:20 2016

Copyright (c) 1982, 2005, Oracle.  All rights reserved.

Enter value for 1: Enter value for 2: Enter value for 3: Connected.
     ad_ctx_ddl.sync_index(l_index);
     *
ERROR at line 96:
ORA-06550: line 96, column 6:
PLS-00201: identifier 'AD_CTX_DDL.SYNC_INDEX' must be declared
ORA-06550: line 96, column 6:
PL/SQL: Statement ignored
```

      - Metalink note followed to fix the above error - 1917695.1

- Copy $AD_TOP/patch/115/sql/adctxpkg.sql from the application server to DB server in the working directory
- Login to the DB (DUMMY) as system and run the script – adctxpkg.sql
  - 1st parameter – system password
  - 2nd parameter – CTXSYS
  - 3rd parameter - APPS

- Run the autoconfig again → this time it completed successfully.



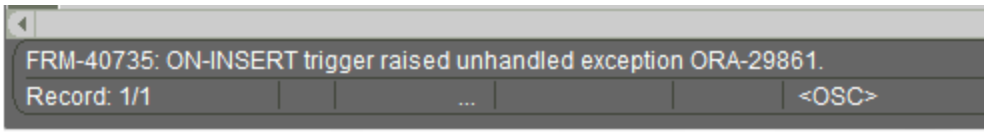
- Start the instance →
  - All the application services came up w/o any error.
  - Able to submit one job which completed successfully and able to view the o/p
  - Support team **reported and issue with the vertex**
    - 

  - While running the PL/SQL block in the DB using XXxxx user, I was getting the same **error**.

```
SQL>set serveroutput on;
SQL>begin
  2  vertexoicquoexample('SALE',1);
  3  end;
  4  /
********************************************************************************
Quotation : SALE request failed with the following error:
Error during sending of XML request document.:ORA-29273: HTTP request failed
ORA-06512: at "SYS.UTL_HTTP", line 1130
ORA-24247: network access denied by access control list (ACL)
********************************************************************************

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.21
SQL>begin
  2  vertexoicquoexample('PURCHASE',1);
  3  end;
  4  /
********************************************************************************
Quotation : PURCHASE request failed with the following error:
Error during sending of XML request document.:ORA-29273: HTTP request failed
ORA-06512: at "SYS.UTL_HTTP", line 1130
ORA-24247: network access denied by access control list (ACL)
********************************************************************************

PL/SQL procedure successfully completed.
```

  - **Troubleshooting**
    - Upon investigating, this looked to be an issue with ACL
    - Ran below SQL to compare the o/p from a working and DUMMY

    - I found few of the ACLs are missing in DUMMY - /sys/acls/utlpkg.xml & /sys/acls/network_services2.xml
    - Created the ACL and assigned it to the required user with the correct privilege
    - Assigned the ACL to the network (**Metalink Document ID - 1080105.1**)
    - Tested the connectivity again by running the same pl/sql block and it completed successfully.
  - 2 spatial indexes were in “FAILED” state and hence causing an error while saving locations.



- Troubleshooting and resolution
  - Ran below query
    - *select owner, index_name, index_type, status, domidx_status, domidx_opstatus from dba_indexes where domidx_opstatus != 'VALID' AND INDEX_NAME NOT IN ('IBC_CITEM_VERSIONS_TL_CT1','TEST_I') order by 1*
  - and it provided 2 INDEXES (PER_ADDRESSES_SPT & HR_LOCATIONS_SPT) OWNED BY HR.
  - Solution is to drop (requery DBA_OBJECTS to confirm if the object is really dropped) and recreate them but while recreating, got an error

```
SQL> CREATE INDEX "HR"."PER_ADDRESSES_SPT" ON "HR"."PER_ADDRESSES" ("GEOMETRY")
   INDEXTYPE IS "MDSYS"."SPATIAL_INDEX"  PARAMETERS ('rebuild_index=MDRT_54B5B$ sdo_indx_dims=2 sdo_rtr_pctfree=10 tablespace=APPS_TS_TX_IDX')  2  ;
CREATE INDEX "HR"."PER_ADDRESSES_SPT" ON "HR"."PER_ADDRESSES" ("GEOMETRY")
*
ERROR at line 1:
ORA-29855: error occurred in the execution of ODCIINDEXCREATE routine
ORA-13203: failed to read USER_SDO_GEOM_METADATA view
ORA-13203: failed to read USER_SDO_GEOM_METADATA view
ORA-06512: at "MDSYS.SDO_INDEX_METHOD_10I", line 10
```

  - Queried "`MDSYS.SDO_GEOM_METADATA_TABLE`" and found that the owner is SYS for the "`sdo_table_name = 'HR_LOCATIONS_ALL'`" & "HR"."PER_ADDRESSES". "OWNER" should be HR.
  - Updated "`MDSYS.SDO_GEOM_METADATA_TABLE`" to set the OWNER as "HR" for these 2 records
    - *update MDSYS.SDO_GEOM_METADATA_TABLE set sdo_owner = 'HR' where sdo_table_name = 'HR_LOCATIONS_ALL'*
  - Recreated the index and it completed successfully

```
SQL> show user
USER is "SYS"
SQL> drop index "HR"."PER_ADDRESSES_SPT";

Index dropped.

SQL> CREATE INDEX "HR"."PER_ADDRESSES_SPT" ON "HR"."PER_ADDRESSES" ("GEOMETRY")
   INDEXTYPE IS "MDSYS"."SPATIAL_INDEX"  PARAMETERS ('rebuild_index=MDRT_54B5B$ sdo_indx_dims=2 sdo_rtr_pctfree=10 tablespace=APPS_TS_TX_IDX')  2  ;

Index created.

SQL> drop index "HR"."HR_LOCATIONS_SPT";

Index dropped.

SQL> CREATE INDEX "HR"."HR_LOCATIONS_SPT" ON "HR"."HR_LOCATIONS_ALL" ("GEOMETRY")
   INDEXTYPE IS "MDSYS"."SPATIAL_INDEX"  PARAMETERS ('rebuild_index=MDRT_54B5C$ sdo_indx_dims=2 sdo_rtr_pctfree=10 tablespace=APPS_TS_TX_IDX')  2  ;

Index created.

SQL>
```

  - Again queried to check the 'domidx_opstatus' from dba_indexes and it was VALID.
- Asked the user to validate the 'save location' functionality and it worked fine.

```
Elapsed: 00:00:00.21
SQL>begin
  2  vertexoicquoexample('SALE',1);
  3  end;
  4  /
********************************************************************************
Quotation : SALE request was successfully processed.
********************************************************************************
--------------------------------------------------------------------
Document Number:
Document Date: 21-APR-04
Transaction ID: T101
Trans Total Amt: 1.8
Line Item Number: 1
Total Tax: 1.8
Tax Type: SALES
Calculated Tax: 1.8
--------------------------------------------------------------------

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.90
SQL>begin
  2  vertexoicquoexample('PURCHASE',1);
  3  end;
  4  /
********************************************************************************
Quotation : PURCHASE request was successfully processed.
********************************************************************************
--------------------------------------------------------------------
Document Number:
Document Date: 21-APR-04
Transaction ID: T101
Trans Total Amt: 1.8
Line Item Number: 1
Total Tax: 1.8
Tax Type: SALES
Calculated Tax: 1.8
--------------------------------------------------------------------

PL/SQL procedure successfully completed.
```