ESRI EXTERNAL

How to Move the Oracle Enterprise Geodatabase with the Oracle Data Pump Utility

community.esri.com

Abstract

This white paper is intended for database administrators as recommendations to move the Oracle Enterprise Geodatabase with the Oracle Data Pump Utility.

Marcelo Marques

Esri Principal Product Engineer

OCP – Oracle Certified Professional

October 2023

Contents

1. Introduction

- The “exp” and “imp” are legacy commands and are desupported by Oracle.
- These commands do not support new advanced Oracle functionality introduced in Oracle 11g and 12c.
- Therefore, is always best to use the oracle data pump “expdp” and “impdp” commands going forward.
- Especially if the database stores Oracle Spatial SDO data.
- Esri also recommends using Oracle data pump expdb/impdb commands with the Geodatabase http://support.esri.com/en/knowledgebase/techarticles/detail/34323
- Because ArcGIS provides a native spatial type for Oracle, ST_Geometry, it leverages many of the advanced functionality in Oracle.
- Advanced functionality such as Oracle's extensible object model, domain indexes, extensible optimizer and the ability to define behavior when the type is exported and imported.
- To ensure the correct behavior when exporting and importing data, ArcGIS recommends the Oracle Data Pump utility is used and not the Oracle Original Export and Import Utility.

2. Oracle Data Pump Documentation

- Oracle 21c Data Pump
- Oracle 19c Data Pump
- Oracle 12.2 Data Pump

3. Oracle Data Pump Best Practices

- Oracle Data Pump Best Practices

3.1. Use Parallelism

Accomplish more work in less time using parallelism. A new Data Pump Job consists of at least two background processes: a Control process and a Worker process, and 2 sessions. The Data Pump PARALLEL parameter creates additional background processes and sessions during an export or import.

The PARALLEL=n parameter specifies the maximum number of processes of active execution operating on behalf of the export or import job. Typically, the value for n should be twice the number of CPU cores but be prepared to adjust it if need be.

3.2. Gather Accurate Statistics Before And After Data Pump Operations

It can be helpful to have accurate statistics prior to an export operation because this helps ensure the best possible export performance. It is also helpful to gather statistics after import.

The frequency of statistics updates that is needed to keep statistics current depends on how volatile the database is. Statistics gathering includes both dictionary statistics and object statistics. Dictionary statistics are used when data pump filters, orders and collects metadata objects at various stages of the export.

Object statistics are used to estimate table and index sizes, which helps produce optimal ordering and parallelism. Concurrent with metadata export, table sizes are estimated and ranked from largest to smallest for parallel export.

The table sizes are estimated using statistics. You can collect statistics using the dbms_stats package, with the gather_table_stats, gather_schema_stats, or gather_database_stats procedure.

Oracle recommends using gather_schema_stats because it will gather stats on all objects regardless of the staleness information.

3.3. Exclude Statistics From Export And Import

Oracle recommends that you do not export statistics during export.

This will provide better performance on both export and import, even accounting for the need to gather statistics after the import. You can exclude statistics from an export operation using the EXCLUDE=STATISTICS parameter or EXCLUDE=TABLE_STATISTICS,INDEX_STATISTICS for Transportable

Tablespaces. Instead, follow the best practice of either creating fresh statistics on the target database or using a DBMS_STATS staging table for transporting statistics.

3.4. Include Diagnostic Parameters During Export And Import

Timestamp the messages that are displayed during an export operation using LOGTIME=ALL. This parameter is available beginning with Oracle Database release 12.1. Having timestamps on every line in the logfile helps when assessing export and import performance. Record the number of objects and the elapsed time of the job in the Oracle Data Pump log file.

Accomplish this using the parameter METRICS=YES. This gives an extra level of detail, such as the work performed by each process in a PARALLEL export or import. Timestamp the messages that are displayed during an import operation using LOGTIME=ALL. This parameter is available beginning with Oracle Database release 12.1. Record the number of objects and the elapsed time for the job in the Oracle Data Pump log file using the parameter METRICS=YES.

3.5. Specify Wildcards To Create Multiple Dumpfiles

When using parallelism, use the %U or %L substitution variable when specifying the dumpfile name. This enables simultaneous parallel writes to multiple logfiles. Also, multiple dump files of a smaller size can be easier to manage and copy.

For example:

For 1 to 99 files use %U (expands the file names into a 2-digit, fixed-width, incrementing integer)

```
dumpfile=dumpfile%U.dmp
filesize=n
```

For 1 to greater than 99 files use %L (expands the file names into a 3-digit to 10-digit, variable-width integer)

```
dumpfile=dumpfile%L.dmp
filesize=n
```

Otherwise, you end up with parallel writes to a single logfile, which can impact performance because each worker writing to the dumpfile would take an exclusive lock on the file, blocking the other worker processes.

3.6. Use Securefile Lobs

Oracle recommends using SecureFile LOBs, especially with partitioning. SecureFile LOBs offer superior performance, functionality and scalability over BasicFile LOBs, including:

- parallel IO into and out of tables with LOB columns
- compression
- encryption

You can use the impdp parameter LOB_STORAGE=SECUREFILE to automatically convert old LOBs to SecureFiles. Tables with SecureFile LOBs storage are automatically created in the target database.

4. Data Pump Export

Use the Oracle Data Pump Export "expdp" command to create the dump file of the Oracle Geodatabase.

- Note: Data Pump Export writes the log file using the database character set. If your client NLS_LANG environment setting sets up a different client character set from the database character set, then it is possible that table names can be different in the log file than they are when displayed on the client output screen. "Oracle 21c Data Pump Export"

- Improve Performance With Parallelism And Current Statistics

4.1. Gather Accurate Statistics

```
CREATE OR REPLACE PROCEDURE PROC_STATS AS
BEGIN
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName          => 'SDE'
    ,Granularity      => 'ALL'
    ,Options          => 'GATHER'
    ,Gather_Temp      => FALSE
    ,Degree           => 4
    ,Cascade          => TRUE
    ,No_Invalidate    => FALSE);
END;
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName          => 'GIS'
    ,Granularity      => 'ALL'
    ,Options          => 'GATHER'
    ,Gather_Temp      => FALSE
    ,Degree           => 8
    ,Cascade          => TRUE
    ,No_Invalidate    => FALSE);
END;
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName          => 'GIS2'
    ,Granularity      => 'ALL'
    ,Options          => 'GATHER'
    ,Gather_Temp      => FALSE
    ,Degree           => 8
    ,Cascade          => TRUE
    ,No_Invalidate    => FALSE);
END;
END;
/

--execute store procedure
BEGIN
  SYS.PROC_STATS;
END;
/
```

4.2. Create the Directory

```
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual;

SELECT directory_name, directory_path FROM dba_directories WHERE
directory_name='DATA_PUMP_DIR';

--CREATE or REPLACE DIRECTORY DATA_PUMP_DIR as
'/home/app/oracle/database/admin/mcs1/dpdump/';

SELECT directory_name, directory_path FROM dba_directories WHERE
directory_name='DATA_PUMP_DIR2';

CREATE or REPLACE DIRECTORY DATA_PUMP_DIR2 as
'/mnt/nfs/redarchive4/db_bkp_act_yes/mcsdboralnx3/mcs1/dpdump/aeropro1';
CREATE or REPLACE DIRECTORY DATA_PUMP_DIR2 as '/tmp';

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO SDE;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS2;
```

4.3. Full data pump export

```
expdp \"sys/*****@ mcsdboralnx3:1575/aeropro1.esri.com as sysdba\"
ACCESS_METHOD=EXTERNAL_TABLE directory=data_pump_dir2 filesize=3G
parallel=8 exclude=statistics logtime=all metrics=yes
dumpfile=dp_aeropro1_full_%U.dmp full=yes logfile=dpexp_aeropro1_full.log
```

4.4. Or Individual Users data pump export

export sde user and each data owner user separate

```
expdp \"sde/*****@mcsdboralnx3:1575/aeropro1.esri.com\"
ACCESS_METHOD=EXTERNAL_TABLE directory=data_pump_dir2
filesize=3G parallel=8 exclude=statistics logtime=all metrics=yes
dumpfile=dp_aeropro1_sde_%U.dmp schemas=sde logfile=dpexp_aeropro1_sde.log

expdp \"gis/*****@mcsdboralnx3:1575/aeropro1.esri.com\"
ACCESS_METHOD=EXTERNAL_TABLE directory=data_pump_dir2
filesize=3G parallel=8 exclude=statistics logtime=all metrics=yes
dumpfile=dp_aeropro1_gis_%U.dmp schemas=gis logfile=dpexp_aeropro1_gis.log

expdp \"gis2/*****@mcsdboralnx3:1575/aeropro1.esri.com\"
ACCESS_METHOD=EXTERNAL_TABLE directory=data_pump_dir2
filesize=3G parallel=8 exclude=statistics logtime=all metrics=yes
dumpfile=dp_aeropro1_gis2_%U.dmp schemas=gis2 logfile=dpexp_aeropro1_gis2.log
```

The data pump "expdp" command splits the export into multiple files, each file up to 3G.

Then zip each file and upload to the ftp site.

Also, upload the expdp ".log" files as well, and include the text file with the output of the queries from "4.6 Database Info Queries".

4.5. Known Issues

```
expdp \"sde/*****@mcsdboralnx3:1575/aeropro1.esri.com\"
ACCESS_METHOD=EXTERNAL_TABLE directory=data_pump_dir2
filesize=3G parallel=8 exclude=statistics logtime=all metrics=yes
dumpfile=dp_aeropro1_sde_%U.dmp schemas=sde logfile=dpexp_aeropro1_sde.log
...
ORA-39001: invalid argument value
ORA-39000: bad dump file specification
ORA-31643: unable to close dump file
"/mnt/smb/redarchive4/pssoldb/db_bkp_act_yes/MCSDBORALNX3/mcs1/dpdump/dp_aeropro
1_sde_01.dmp"
ORA-19510: failed to set size of 1 blocks for file
"/mnt/smb/redarchive4/pssoldb/db_bkp_act_yes/MCSDBORALNX3/mcs1/dpdump/dp_aeropro
1_sde_01.dmp" (block size=4096)
ORA-27086: unable to lock file - already in use
Linux-x86_64 Error: 13: Permission denied
Additional information: 8
Additional information: 23652

--MOS: RMAN and/or Datapump Export Backup fails with ORA-27086 on NFS / NAS (Doc ID
1187014.1)
Workaround: skip to export to the windows samba share and
            export to /tmp or to a linux nfs share
```

4.6. Database Info Queries

```
connect /as sysdba
spool db_info.txt

-- version
select * from v$version;

-- NLS parameters
select value from NLS_DATABASE_PARAMETERS where parameter =
'NLS_NCHAR_CHARACTERSET';
select * from v$nls_parameters;

-- Geodatabase version
select * from sde.version;

--User-schema geodatabases;
select * from sde.instances;

--geodatabase data owners
select distinct(owner) from sde.table_registry order by 1;

--database users
select distinct(username) from all_users order by 1;

--database tablepaces per user
select owner,tablespace_name,'tb'
from all_tables
group by owner,owner,tablespace_name
union all
select owner,tablespace_name,'idx'
from all_indexes
group by owner,owner,tablespace_name
order by 1,2;

spool off;
exit;
```

Also, upload the “db_info.txt” file to the ftp site

4.7. Platform Information

Also add the Platform Information in the "db_info.txt" file.

a. Database Server OS Version

Example: Oracle Exadata X5 running Oracle Linux 6 Update 6 with the Unbreakable Enterprise Kernel 2

b. Oracle Database release

Example: Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

c. Geodatabase Spatial Storage Type

Examples: Esri ST_GEOMETRY, Esri SDELOB, Oracle SDO_GEOMETRY

d. Oracle Database Release and Patch Version – Opatch

```
su - oracle
cd $ORACLE_HOME
cd OPatch
./opatch lsinventory
```

Also, upload the opatch output ".txt" file to the ftp site.

Note: This will assist to import the data using data pump import, “impdp” command in the target database.

----

5. Data Pump Import

Use the Oracle Data Pump Import "impdp" command to import the dump file of the Oracle Geodatabase.

5.1. Oracle SID and PDB Name

- you can import into a database with a different SID than the source, CDB or No-CDB
- you can import into a pluggable database with a different name than the source, if using a CDB
- This will not break the geodatabase

5.2. NLS Character Settings

a. Preferably the target database shall have the same NLS * settings as the source database.

b. NLS = National Language Support

c. Oracle Data Pump Import writes the log file using the database character set. If your client NLS_LANG environment sets up a different client character set from the database character set, then it is possible that table names can be different in the log file than they are when displayed on the client output screen. "Oracle 21c Data Pump Import".

d. Migrating Character Data Using a Full Export and Import: a full export and import can also be used to convert the database to a new character set. It may be more time-consuming and resource-intensive as a separate target instance must be set up. If you plan to migrate your data to a non-Unicode character set, which Oracle strongly discourages, you can use the DMU * to look for invalid character representation issues in the database and use export and import for the data conversion. Note that the DMU * will not correctly identify data expansion issues (column and data type limit violations) if the migration is not to Unicode. It will also not identify characters that exist in the source database character set but do not exist in the non-Unicode target character set. "Oracle Character Set Conversion Issues".

   * Database Migration Assistant for Unicode (DMU) software

5.3. Oracle Support Articles

Oracle Support Document 175624.1 (Oracle Server - Export and Import FAQ)
https://support.oracle.com/epmos/faces/DocumentDisplay?id=175624.1

Oracle Support Document 227332.1 (NLS considerations in Import/Export - Frequently Asked Questions)
https://support.oracle.com/epmos/faces/DocumentDisplay?id=227332.1

How are expdp and impdp affected by the NLS_LANGUAGE and NLS_TERRITORY ?

Not. Normally you use the AMERICAN_AMERICA default, but if you imported with NLS_LANG set to FRENCH_FRANCE for example then you will not have problems, even if the originating environment used GERMAN_GERMANY or so.

There is no relation between NLS_LANGUAGE and NLS_TERRITORY and the actual languages stored in the database.

How does NLS_LANG affect DataPump (expdp/impdp)?

Datapump does **not** use the NLS_LANG to do conversion between databases.
Conversion between 2 database charactersets is done purely based on the NLS_CHARACTERSET (or NLS_NCHAR_CHARACTERSET for Nchar,Nvarchar and Nclob datatypes) of the source and target database.

However, if you specify a parameter file then the NLS_LANG *is* used for the encoding of the parameter file. This is only important if you use non-English characters (e.g. for the QUERY parameter) in the parameter file.
If you use non-English characters in the parameter file then the NLS_LANG environment variable should be set (in the session where the Data Pump job is started) to the correct encoding of the parameter file (!).

Do **NOT** use Expdp/Impdp when going to (AL32)UTF8 or an other multibyte characterset on **ALL** 10g versions lower than 10.2.0.4 (including 10.1.0.5). Also 11.1.0.6 is affected.
It will provoke data corruption *unless* Patch 5874989 is applied on the Impdp side. Expdp is not affected, hence the data in the dump file is correct. Also the "old" exp/imp tools are not affected.
This problem is fixed in the 10.2.0.4 and 11.1.0.7 patch set.
Fixed in 11.2.0.1 and up
For windows the fix is included in
10.1.0.5.0 Patch 20 (10.1.0.5.20P) or later, see Note 276548.1 .
10.2.0.3.0 Patch 11 (10.2.0.3.11P) or later, see Note 342443.1 .

What causes ORA-01401 or ORA-12899 during import (imp and impdp) ?

9i and lower gives ORA-01401: inserted value too large for column , 10g and up gives ORA-12899: value too large for column

This may be seen when exporting from a database with a 8 bit NLS_CHARACTERSET like WE8ISO8859P1, WE8MSWIN1252 , WE8DEC ...) to a database with a 16 bit NLS_CHARACTERSET (like JA16SJIS , ZHS16GBK, KO16MSWIN949) or NLS_CHARACTERSET set to AL32UTF8 or UTF8
Please see Note 1297961.1 ORA-01401 / ORA-12899 While Importing Data In An AL32UTF8 / UTF8 (Unicode) Or Other Multibyte NLS_CHARACTERSET Database.

6. Prepare Geodatabase for Data Pump Import

6.1. Pluggable Database Enable Force No Logging

- If you are using a CDB then you can “disable archivelog” or set the PDB to “force no logging” to improve performance to import a large dump file

- If you have a NO-CDB then you can “disable archivelog”

- another option if you cannot “disable archivelog” is to use the data pump import parameter TRANSFORM=DISABLE_ARCHIVE_LOGGING:Y

a. Disable Archivelog

```
--Turn off Archive

--Windows
set oracle_sid=mcs

--Linux
.oraenv
mcs

------------------------------
sqlplus /nolog
connect /as sysdba
------------------------------
shutdown immediate;
startup mount;
alter database flashback off;
alter database noarchivelog;
select flashback_on from v$database;
alter database open;
select host_name,instance_name,status,database_status from v$instance;
select name,open_mode,log_mode,flashback_on from v$database;
```

b. Enable PDB Force No Logging

```
sqlplus /nolog
connect sys /as sysdba
----------------------------------------------------------------
--enable pluggable database force nologging
ALTER SESSION SET CONTAINER = cdb$root;
show con_name
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE aeropro1 OPEN RESTRICTED;
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 ENABLE FORCE NOLOGGING;
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE aeropro1 OPEN READ WRITE;
SELECT * FROM dba_pdbs ORDER BY pdb_name;
SELECT * FROM v$pdbs ORDER BY name;
----------------------------------------------------------------
--RAC
--enable pluggable database force nologging
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE INSTANCES=ALL;
--open restrictted pdb only on rac node 1
ALTER PLUGGABLE DATABASE aeropro1 OPEN RESTRICTED INSTANCES = ('mcs11');
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 ENABLE FORCE NOLOGGING;
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE INSTANCES=ALL;
ALTER PLUGGABLE DATABASE aeropro1 OPEN READ WRITE INSTANCES=ALL;
SELECT * FROM dba_pdbs ORDER BY pdb_name;
SELECT * FROM v$pdbs ORDER BY name;
----------------------------------------------------------------
```

6.2. Create Tablespaces

```
------------------------------------------------------------------------------------------------------------------------------- -------------
DROP TABLESPACE GIS_SDE INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE GIS_SDEINDEX INCLUDING CONTENTS AND DATAFILES;

CREATE BIGFILE TABLESPACE GIS_SDE
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE BIGFILE TABLESPACE GIS_SDEINDEX
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 100M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
------------------------------------------------------------------------------------------------------------------------------- -------------
DROP TABLESPACE GIS_DATA INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE GIS_INDEX INCLUDING CONTENTS AND DATAFILES;

CREATE BIGFILE TABLESPACE "GIS_DATA"
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE BIGFILE TABLESPACE "GIS_INDEX"
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
------------------------------------------------------------------------------------------------------------------------------- -------------
DROP TABLESPACE GIS2_DATA INCLUDING CONTENTS AND DATAFILES;
DROP TABLESPACE GIS2_INDEX INCLUDING CONTENTS AND DATAFILES;

CREATE BIGFILE TABLESPACE "GIS2_DATA"
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';

CREATE BIGFILE TABLESPACE "GIS2_INDEX"
DATAFILE '+DATA' SIZE 10M AUTOEXTEND ON NEXT 1M MAXSIZE 400M
LOGGING EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K
SEGMENT SPACE MANAGEMENT AUTO
DEFAULT COMPRESS FOR OLTP STORAGE ( ENCRYPT ) ENCRYPTION USING 'AES256';
```

6.3. Tablespaces No Logging

- Be aware that if you are performing a Data Pump Import into a table or tablespace created with the NOLOGGING clause enabled, then a redo log file may still be generated. The redo that is generated in such a case is generally for maintenance of the Data Pump control table, or related to underlying recursive space transactions, data dictionary changes, and index maintenance for indices on the table that require logging.

```
/*
Disable Tablespace logging

Yes - Generate redo logs for creation of tables, indexes and partitions, and for subsequent inserts. Recoverable

No - Redo log entries are smaller; the above operations are not logged and not recoverable.

*/

sqlplus /nolog
connect / as sysdba;

SELECT owner , table_name
FROM dba_tables
WHERE logging='NO';

SELECT owner , table_name, index_name
FROM dba_indexes
WHERE logging='YES';

SELECT tablespace_name
FROM dba_tablespaces
WHERE logging='LOGGING';

SELECT TABLE_NAME, PARTITION_NAME
FROM user_tab_partitions
WHERE logging='YES';

SELECT 'ALTER TABLESPACE ' || CHR(34) || NAME || CHR(34) || ' NOLOGGING;'
  FROM V$TABLESPACE
  WHERE NAME NOT IN
  ('SYSTEM','SYSAUX','USERS','TEMP','TEMP2','TEMP3','TEMP4','UNDOTBS1')
ORDER BY NAME;

ALTER TABLESPACE "GIS_SDE" NOLOGGING;
ALTER TABLESPACE "GIS_SDEINDEX" NOLOGGING;
ALTER TABLESPACE "GIS_DATA" NOLOGGING;
ALTER TABLESPACE "GIS_INDEX" NOLOGGING;
ALTER TABLESPACE "GIS2_DATA" NOLOGGING;
ALTER TABLESPACE "GIS2_INDEX" NOLOGGING;
```

6.4. Recreate Profile GIS Users

```
CREATE PROFILE "GIS_USERS"
LIMIT CPU_PER_SESSION DEFAULT
CPU_PER_CALL DEFAULT
CONNECT_TIME DEFAULT
IDLE_TIME 60                                    <== 60 minutes
SESSIONS_PER_USER DEFAULT
LOGICAL_READS_PER_SESSION DEFAULT
LOGICAL_READS_PER_CALL DEFAULT
PRIVATE_SGA DEFAULT
COMPOSITE_LIMIT DEFAULT
PASSWORD_LIFE_TIME UNLIMITED
PASSWORD_GRACE_TIME DEFAULT
PASSWORD_REUSE_MAX DEFAULT
PASSWORD_REUSE_TIME DEFAULT
PASSWORD_LOCK_TIME DEFAULT
FAILED_LOGIN_ATTEMPTS DEFAULT
PASSWORD_VERIFY_FUNCTION DEFAULT;

ALTER PROFILE "GIS_USERS"
LIMIT IDLE_TIME 1440                                    <== 1440 minutes = 24 hours
PASSWORD_LIFE_TIME UNLIMITED
PASSWORD_GRACE_TIME UNLIMITED;

ALTER PROFILE "DEFAULT" LIMIT IDLE_TIME UNLIMITED
PASSWORD_LIFE_TIME UNLIMITED
PASSWORD_GRACE_TIME UNLIMITED;

SELECT * FROM DBA_PROFILES ORDER BY PROFILE;
```

6.5. Recreate Temp Tablespace Group

```
CREATE SMALLFILE TEMPORARY TABLESPACE "TEMP2"
TEMPFILE '+DATA(TEMPFILE)' SIZE 2G
AUTOEXTEND ON NEXT 640K MAXSIZE 2G
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1024K;

CREATE SMALLFILE TEMPORARY TABLESPACE "TEMP3"
TEMPFILE '+DATA(TEMPFILE)' SIZE 2G
AUTOEXTEND ON NEXT 640K MAXSIZE 2G
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1024K;

CREATE SMALLFILE TEMPORARY TABLESPACE "TEMP4"
TEMPFILE '+DATA(TEMPFILE)' SIZE 2G
AUTOEXTEND ON NEXT 640K MAXSIZE 2G
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1024K;


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

ALTER TABLESPACE TEMP  TABLESPACE GROUP DEFAULT_TEMP_GROUP;
ALTER TABLESPACE TEMP2 TABLESPACE GROUP DEFAULT_TEMP_GROUP;

ALTER DATABASE DEFAULT TEMPORARY TABLESPACE DEFAULT_TEMP_GROUP;

ALTER TABLESPACE TEMP3 TABLESPACE GROUP GIS_TEMP_GROUP;
ALTER TABLESPACE TEMP4 TABLESPACE GROUP GIS_TEMP_GROUP;


------------------------------------------------------------------------------------------------------------------------------------------------------------
/*
SELECT 'ALTER DATABASE TEMPFILE ' || CHR(39) || name || CHR(39) || ' RESIZE 2G;' AS SQLTXT
FROM V$TEMPFILE
UNION ALL
SELECT 'ALTER DATABASE TEMPFILE ' || CHR(39) || name || CHR(39) || ' AUTOEXTEND ON MAXSIZE
2G;' AS SQLTXT
FROM V$TEMPFILE
ORDER BY SQLTXT;
*/
---------------------------------------------------------------------------------------------------------------------------------------------
ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP.DBF' AUTOEXTEND ON MAXSIZE 2G;
ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP.DBF' RESIZE 2G;

ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP2.DBF' AUTOEXTEND ON MAXSIZE 2G;
ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP2.DBF' RESIZE 2G;

ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP3.DBF' AUTOEXTEND ON MAXSIZE 2G;
ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP3.DBF' RESIZE 2G;

ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP4.DBF' AUTOEXTEND ON MAXSIZE 2G;
ALTER DATABASE TEMPFILE '+DATA/MCS/AEROPRO1/TEMPFILE/TEMP4.DBF' RESIZE 2G;
---------------------------------------------------------------------------------------------------------------------------------------------
```

6.6. Recreate SDE user

```
GRANT EXECUTE ON DBMS_PIPE TO public;
GRANT EXECUTE ON DBMS_LOCK TO public;
GRANT EXECUTE ON DBMS_UTILITY TO public;
GRANT EXECUTE ON DBMS_SQL TO public;
GRANT EXECUTE ON DBMS_LOB TO public;
GRANT EXECUTE ON UTL_RAW TO public;

DROP USER SDE CASCADE;

CREATE USER SDE PROFILE "GIS_USERS" IDENTIFIED BY "*****"
DEFAULT TABLESPACE GIS_SDEDICT TEMPORARY TABLESPACE TEMP;

ALTER USER "SDE" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";

ALTER USER "SDE" PROFILE "GIS_USERS";

ALTER USER "SDE" QUOTA UNLIMITED ON "GIS_SDE";
ALTER USER "SDE" QUOTA UNLIMITED ON "GIS_SDEINDEX";

ALTER USER SDE DEFAULT TABLESPACE GIS_SDE;

CREATE ROLE GIS_SDE_MASTER;

GRANT CREATE SESSION,
  CREATE TABLE,
  CREATE VIEW,
  CREATE PROCEDURE,
  CREATE SEQUENCE,
  CREATE TRIGGER,
  CREATE TYPE,
  CREATE INDEXTYPE,
  CREATE LIBRARY,
  CREATE PUBLIC SYNONYM,
  DROP PUBLIC SYNONYM,
  ADMINISTER DATABASE TRIGGER,
  CREATE OPERATOR
TO GIS_SDE_MASTER;

GRANT SELECT_CATALOG_ROLE TO SDE;
GRANT SELECT_CATALOG_ROLE TO GIS_SDE_MASTER;

GRANT EXECUTE ON DBMS_CRYPTO TO GIS_SDE_MASTER;

GRANT EXECUTE ON DBMS_CRYPTO TO SDE;
GRANT EXECUTE ON DBMS_PIPE TO SDE;

GRANT GIS_SDE_MASTER TO SDE;

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO SDE;
```

6.7. Recreate GIS data owner role

```
--CREATE ROLE GIS_LOGIN
DROP ROLE "GIS_LOGIN";
CREATE ROLE "GIS_LOGIN"  NOT IDENTIFIED;
GRANT CREATE SESSION TO "GIS_LOGIN";
GRANT CREATE TABLE TO "GIS_LOGIN";
GRANT CREATE SEQUENCE TO "GIS_LOGIN";

--CREATE ROLE GIS_DATA_OWNER
DROP ROLE "GIS_DATA_OWNER";
CREATE ROLE "GIS_DATA_OWNER"  NOT IDENTIFIED;
GRANT CREATE SESSION TO "GIS_DATA_OWNER";
GRANT CREATE SEQUENCE TO "GIS_DATA_OWNER";
GRANT CREATE TRIGGER TO "GIS_DATA_OWNER";
GRANT CREATE VIEW TO "GIS_DATA_OWNER";
GRANT CREATE TABLE TO "GIS_DATA_OWNER";
GRANT CREATE PROCEDURE TO "GIS_DATA_OWNER";
```

6.8. Recreate GIS data owner user

```
DROP USER GIS CASCADE;

CREATE USER GIS PROFILE "GIS_USERS" IDENTIFIED BY "*****"
DEFAULT TABLESPACE GIS_DATA TEMPORARY TABLESPACE TEMP;

ALTER USER "GIS" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";

ALTER USER "GIS" PROFILE "GIS_USERS";

GRANT GIS_DATA_OWNER TO GIS;

ALTER USER "GIS" QUOTA UNLIMITED ON "GIS_DATA"
  DEFAULT TABLESPACE "GIS_DATA"
  QUOTA UNLIMITED ON "GIS_INDEX"
  ACCOUNT UNLOCK;

ALTER USER GIS DEFAULT TABLESPACE GIS_DATA;

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS;
```

6.9. Recreate GIS2 data owner user

```
DROP USER GIS2 CASCADE;

CREATE USER GIS2 PROFILE "GIS_USERS" IDENTIFIED BY "*****"
DEFAULT TABLESPACE GIS2_DATA TEMPORARY TABLESPACE TEMP;

ALTER USER "GIS2" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";

ALTER USER "GIS2" PROFILE "GIS_USERS";

GRANT GIS_DATA_OWNER TO GIS2;

ALTER USER "GIS2" QUOTA UNLIMITED ON "GIS2_DATA"
  DEFAULT TABLESPACE "GIS2_DATA"
  QUOTA UNLIMITED ON "GIS2_INDEX"
  ACCOUNT UNLOCK;

ALTER USER GIS2 DEFAULT TABLESPACE GIS2_DATA;

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS2;
```

6.10. Resize Tablespaces

You need to resize the tablespaces to avoid fragmentation and to make the data pump import process to perform faster and avoid tablespace autoextend when you are importing a large dump file.

```
SELECT * FROM V$DATAFILE;
------------------------------------------------------------------------------------------------------------------------------------------
SELECT 'ALTER DATABASE DATAFILE ' || CHR(39) || name || CHR(39) || ' RESIZE 50G;' AS SQLTXT
FROM V$DATAFILE
UNION ALL
SELECT 'ALTER DATABASE DATAFILE ' || CHR(39) || name || CHR(39) || ' AUTOEXTEND ON MAXSIZE
50G;' AS SQLTXT
FROM V$DATAFILE
ORDER BY SQLTXT;
------------------------------------------------------------------------------------------------------------------------------------------
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_SDE.DBF'
AUTOEXTEND ON MAXSIZE 10G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_SDEINDEX.DBF'
AUTOEXTEND ON MAXSIZE 10G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_SDE.DBF'
RESIZE 10G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_SDEINDEX.DBF'
RESIZE 10G;
------------------------------------------------------------------------------------------------------------------------------------------
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_DATA.DBF'
AUTOEXTEND ON MAXSIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_INDEX.DBF'
AUTOEXTEND ON MAXSIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_DATA.DBF'
RESIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS_INDEX.DBF'
RESIZE 50G;
------------------------------------------------------------------------------------------------------------------------------------------
ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS2_DATA.DBF'
AUTOEXTEND ON MAXSIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS2_INDEX.DBF'
AUTOEXTEND ON MAXSIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS2_DATA.DBF'
RESIZE 50G;

ALTER DATABASE DATAFILE 'D:\ORADATA\MCS1\AEROPRO1\DATAFILE\GIS2_INDEX.DBF'
RESIZE 50G;
------------------------------------------------------------------------------------------------------------------------------------------
```

6.11. Data Pump Import

Use the Oracle Data Pump Import “impdp” command to import the dump file of the Oracle Geodatabase.

a. Create the Directory

```
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual;

SELECT directory_name, directory_path FROM dba_directories WHERE
directory_name='DATA_PUMP_DIR';

--CREATE or REPLACE DIRECTORY DATA_PUMP_DIR as
'/home/app/oracle/database/admin/mcs1/dpdump/';

SELECT directory_name, directory_path FROM dba_directories WHERE
directory_name='DATA_PUMP_DIR2';

CREATE or REPLACE DIRECTORY DATA_PUMP_DIR2 as
'/mnt/nfs/redarchive4/db_bkp_act_yes/mcsdboralnx7/mcs1/dpdump/aeropro1';
CREATE or REPLACE DIRECTORY DATA_PUMP_DIR2 as '/tmp';

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO SDE;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS2;
```

b. Known Issues Esri Support

```
impdp \"sde/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=sde content=all
logfile=data_pump_dir2:dpimp_aeropro1_sde.log parallel=4
remap_tablespace=sde:gis_sde,sdeindex:gis_sdeindex
transform=disable_archive_logging:y
----------------------------------------------------------------------------------------------------------------------------------------
--dpimp_aeropro1_sde.log--
Import: Release 19.0.0.0.0 - Production on Tue Sep 13 10:03:36 2022
Version 19.16.0.0.0

Copyright (c) 1982, 2022, Oracle and/or its affiliates.  All rights reserved.
;;;
Connected to: Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production
ORA-39000: bad dump file specification
ORA-39155: error expanding dump file name
"\\redarchive4\pssoldb\db_bkp_act_yes\MCSDBORA3\mcs1\dpdump\aeropro1\dp_aeropro1_sde_01.
dmp"
ORA-48128: opening of a symbolic link is disallowed
----------------------------------------------------------------------------------------------------------------------------------------
--MOS: DataPump Export (EXPDP) ORA-nnnnn Errors (Doc ID 1941688.1)
Workaround: copy dump files from windows network share folder to oracle server
E:\temp\dpdump\aeropro1
----------------------------------------------------------------------------------------------------------------------------------------
```

```
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
SELECT SYS_CONTEXT('USERENV', 'CON_NAME') FROM dual;

CREATE or REPLACE DIRECTORY DATA_PUMP_DIR2 as 'E:/temp/dpdump/aeropro1';

GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO SDE;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS;
GRANT READ, WRITE ON DIRECTORY DATA_PUMP_DIR2 TO GIS2;
```

For more information see at the end of the document “10. Esri Support Articles”.

c. Known Issues Oracle Support

MOS - My Oracle Support

Bug 34409391 - ORA-43853 Error During Data Pump Import With Non-ASSM Target Tablespace and TRANSFORM=LOB_STORAGE:SECUREFILE (**Doc ID 34409391.8**)

Note that this fix can cause / expose the problem described in **Bug:34677088**

Note that this fix has been superseded by the fix in **Bug:34677088**

Bug 34677088 - Data Pump Import Raises ORA-31602 While Setting a Transform Parameter For an IDENTITY_COLUMN Object Type (**Doc ID 34677088.8**)

MOS: EXCLUDE=STATISTICS Or EXCLUDE=INDEX_STATISTICS During Datapump Import Still Analyzes The Indexes (**Doc ID 793585.1**)

SYMPTOMS
You are using Data Pump import (impdp) using the following parameters:

```
EXCLUDE=STATISTICS

- OR -

EXCLUDE=INDEX_STATISTICS
EXCLUDE=TABLE_STATISTICS
```

Tables are not being analyzed in both cases, however, it is still analyzing the indexes.

CAUSE
Oracle, by default, collects statistics for an index during index creation. It is done by design.
The internal parameter "_optimizer_compute_index_stats", is set to TRUE by default.

SOLUTION
This parameter can be set to FALSE to avoid the index statistics during import.

EXAMPLE:

```
SQL> alter system set "_optimizer_compute_index_stats"=FALSE;
```

- OR -

Set the parameter in the pfile/spfile

```
_optimizer_compute_index_stats=FALSE
```

7. Import the SDE user

Note: !!! The SDE user must be imported first !!!

7.1. Import from Full Export

```
--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
impdp \"sde/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp
schemas=sde SQLFILE=data_pump_dir2:dpexp_aeropro1_sde.sql
logtime=all metrics=yes
```

Note: Data Pump Import SQL FILE parameter specifies a file into which all the SQL DDL that Import prepares to execute is written, based on other Import parameters selected.

```
impdp \"sde/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=sde content=all
logfile=data_pump_dir2:dpimp_aeropro1_sde.log
parallel=4 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=sde:gis_sde,sdeindex:gis_sdeindex
```

7.2. Geodatabase SDE dbtune

Note: if you need to use the impdp **remap_tablespace** clause to change the tablespaces then after the sde user and all the data owner users are imported you need to check the tablespace names in the sde.dbtune table.

```
select * from sde.dbtune order by 1,2;
```

It might be necessary to export the sde.dbtune configuration using ArcCatalog / ArcGIS Pro - GP tools, then edit the tablespace names in the exported dbtune text file and reimport the dbtune file into the Geodatabase. For more information visit the links below.

Configuration keywords

Alter configuration keywords

7.3. Or Import from individual sde user export file

```
--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
\"sde/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_sde_%U.dmp SQLFILE=data_pump_dir2:dpexp_aeropro1_sde.sql
logtime=all metrics=yes
```

```
impdp \"sde/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_sde_%U.dmp full=y content=all
logfile=data_pump_dir2:dpimp_aeropro1_sde.log
parallel=4 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=sde:gis_sde,sdeindex:gis_sdeindex
```

This message during the import can be ignored.

```
ORA-31684: Object type INDEX:"SDE"."A1_IX1" already exists
```

Note: Error: ORA-31684 'Index already exists' when importing ST_GEOMETRY spatial indexes using Oracle data pump utility (IMPDP).

https://support.esri.com/en/technical-article/000012233

7.4. Fix invalid objects

Fix invalid object and only then continue to import the other data owner users.

```
sqlplus /nolog
connect /as sysdba
alter session set container = aeropro1;
show con_name;
select count(*), status from dba_objects where status != 'VALID' group by status;
select * from dba_objects where status != 'VALID';

select 'CREATE OR REPLACE PUBLIC SYNONYM ' || object_name || ' FOR SDE.' ||  object_name || ';'
  from dba_objects where status != 'VALID' and object_type = 'SYNONYM' and object_name not like
'DBA%';
/*
CREATE OR REPLACE PUBLIC SYNONYM USER_ST_GEOMETRY_COLUMNS FOR
SDE.USER_ST_GEOMETRY_COLUMNS;
CREATE OR REPLACE PUBLIC SYNONYM ALL_ST_GEOMETRY_COLUMNS FOR
SDE.ALL_ST_GEOMETRY_COLUMNS;
CREATE OR REPLACE PUBLIC SYNONYM USER_ST_GEOM_INDEX FOR SDE.USER_ST_GEOM_INDEX;
*/
EXEC dbms_utility.compile_schema( 'SDE', compile_all => FALSE );
```

7.5. Check Invalid Indexes

```
alter session set container = aeropro1
show con_name
select * from all_indexes d where d.status not in ('VALID','N/A');
```

7.6. Verify sde schema storage

```
sqlplus /nolog
connect sde/****@aeropro1.esri.com
show con_name

--tables
SELECT TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_TABLES
ORDER BY 1;

SELECT TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_TABLES
WHERE STATUS <> 'VALID'
ORDER BY 1;

--indexes
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_INDEXES
ORDER BY 2,1;

SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_INDEXES
WHERE STATUS <> 'VALID'
ORDER BY 1;
```

7.7. Change sde schema storage

If tables and indexes are in the same tablespace then move them to separate tablespaces to reduce datafile fragmentation and to reduce datafile contention and to improve performance, this also improves the performance of the rebuild indexes maintenance task.

```
sqlplus /nolog
connect sde/****@aeropro1.esri.com
show con_name

--move tables
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS_SDE;' AS SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = 'GIS_SDEINDEX'
UNION
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS_SDEDICT;' AS
SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = '' OR TABLESPACE_NAME IS NULL
ORDER BY 1;

--move indexes
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE
GIS_SDEINDEX;' AS SQLMOVE
  FROM USER_INDEXES
    WHERE TABLESPACE_NAME = 'GIS_SDE' AND INDEX_TYPE = 'NORMAL' AND
TEMPORARY = 'N'
UNION
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE
GIS_SDEINDEX;' AS SQLMOVE
  FROM USER_INDEXES
    WHERE (TABLESPACE_NAME = '' OR TABLESPACE_NAME IS NULL) AND INDEX_TYPE =
'NORMAL' AND TEMPORARY = 'N'
ORDER BY 1;

--MOS: ALTER INDEX REBUILD ONLINE Fails with ORA-01450 (Doc ID 236329.1)
--Solution: do not use the “online” clause to rebuild the index
```

Then, verify invalid objects and invalid indexes again.

Note: read “7.2 Geodatabase SDE dbtune“.

7.8. Update the ST_GEOMETRY library

Copy the version of the ST_GEOMETRY library to the Oracle Database Server.

For more information read - Configure extproc to access ST_Geometry in Oracle

a. Update the sde.st_shapelib library if the path changed.

```
SQL> ALTER SESSION SET CONTAINER = aeropro1;
SQL> SHOW CON_NAME

SQL> CREATE OR REPLACE LIBRARY sde.st_shapelib
         AS '/orahome1/dbbase1/esrilibs/pro_30/libst_shapelib.so';
```

b. Check the library

```
SELECT * FROM ALL_LIBRARIES WHERE OWNER = 'SDE' ORDER BY OWNER, LIBRARY_NAME;
```

c. Check Invalid Objects

```
SQL>  SELECT * FROM DBA_OBJECTS
          WHERE STATUS!= 'VALID'
          ORDER BY OWNER, OBJECT_TYPE, OBJECT_NAME;
```

d. Recompile Invalid Objects and then check again for invalid objects.

```
SQL> EXEC dbms_utility.compile_schema( 'SDE', compile_all => FALSE );
```

e. Validate the extproc configuration

Note: connect as the sde user or as the data owner user !!!

```
SQL> SELECT sde.st_geometry(1,1,1,1,0) FROM dual;
SQL> SELECT sde.st_geometry('point (1 1)', 0) FROM dual;
SQL> SELECT sde.ST_AsText(SDE.ST_Geometry('POINT (10 10)', 0)) FROM dual;
```

8. Import the Data Owner Users

8.1. Import from Full Export

```
--GIS Data Owner—

--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis SQLFILE=data_pump_dir2:dpimp_aeropro1_gis.sql
logtime=all metrics=yes

impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis content=all
logfile=data_pump_dir2:dpimp_aeropro1_gis.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data:gis_data,aero_index:gis_index

--EXCLUDE=GRANT – can exclude the grants and refresh roles permissions after
impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis content=all exclude=grant
logfile=data_pump_dir2:dpimp_aeropro1_gis.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data:gis_data,aero_index:gis_index
```

```
--GIS2 Data Owner—

--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
impdp \"gis2/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis2 SQLFILE=data_pump_dir2:dpimp_aeropro1_gis2.sql
logtime=all metrics=yes

impdp \"gis2/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis2 content=all
logfile=data_pump_dir2:dpimp_aeropro1_gis2.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data2:gis2_data,aero_index2:gis2_index

--EXCLUDE=GRANT – can exclude the grants and refresh roles permissions after
impdp \"gis2/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_full_%U.dmp schemas=gis2 content=all exclude=grant
logfile=data_pump_dir2:dpimp_aeropro1_gis2.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data2:gis2_data,aero_index2:gis2_index
```

Note: Data Pump Import SQL FILE parameter specifies a file into which all the SQL DDL that Import prepares to execute is written, based on other Import parameters selected.

8.2. Or Import from individual export file

```
--GIS Data Owner—

--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis_%U.dmp SQLFILE=data_pump_dir:dpexp_aeropro1_gis.sql parallel=8
logtime=all metrics=yes

impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis_%U.dmp full=y content=all logfile=data_pump_dir2:dpimp_aeropro1_gis.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data:gis_data,aero_index:gis_index

--EXCLUDE=GRANT – can exclude the grants and refresh roles permissions after
impdp \"gis/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis_%U.dmp full=y content=all exclude=grant
logfile=data_pump_dir2:dpimp_aeropro1_gis.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data:gis_data,aero_index:gis_index
```

```
--GIS2 Data Owner—

--SQLFILE – show dump file SQL DDL statements – useful during troubleshooting problems
impdp \"gis2/****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis2_%U.dmp SQLFILE=data_pump_dir:dpexp_aeropro1_gis2.sql parallel=8
logtime=all metrics=yes

impdp \"gis2/****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis2_%U.dmp full=y content=all
logfile=data_pump_dir2:dpimp_aeropro1_gis2.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data2:gis2_data,aero_index2:gis2_index

--EXCLUDE=GRANT – can exclude the grants and refresh roles permissions after
impdp \"gis2/*****@mcsdboralnx7:1577/aeropro1.esri.com\" directory=data_pump_dir2
dumpfile=dp_aeropro1_gis2_%U.dmp full=y content=all exclude=grant
logfile=data_pump_dir2:dpimp_aeropro1_gis2.log
parallel=8 exclude=statistics logtime=all metrics=yes
transform=disable_archive_logging:y,lob_storage:securefile
remap_tablespace=aero_data:gis2_data,aero_index2:gis2_index
```

Note: Data Pump Import SQL FILE parameter specifies a file into which all the SQL DDL that Import prepares to execute is written, based on other Import parameters selected.

8.3. Fix invalid objects

```
sqlplus /nolog
connect /as sysdba
alter session set container = aeropro1;
show con_name;
select count(*), status from dba_objects where status != 'VALID' group by status;
select * from dba_objects where status != 'VALID';

EXEC dbms_utility.compile_schema( 'GIS', compile_all => FALSE );
EXEC dbms_utility.compile_schema( 'GIS2', compile_all => FALSE );
```

8.4. Check Invalid Indexes

```
alter session set container = aeropro1
show con_name
select * from all_indexes d where d.status not in ('VALID','N/A');
```

This message during the import can be ignored.

```
ORA-31684: Object type INDEX:"GIS"."A377_IX1" already exists
```

Note: Error: ORA-31684 'Index already exists' when importing ST_GEOMETRY spatial indexes using Oracle data pump utility (IMPDP).

https://support.esri.com/en/technical-article/000012233

8.5. Verify data owner schema storage

```
sqlplus /nolog
connect gis/****@aeropro1.esri.com
--or
connect gis2/****@aeropro1.esri.com

show con_name

--tables
SELECT TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_TABLES
ORDER BY 1;

SELECT TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_TABLES
WHERE STATUS <> 'VALID'
ORDER BY 1;

--indexes
SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_INDEXES
ORDER BY 2,1;

SELECT INDEX_NAME, TABLE_NAME, TABLESPACE_NAME, STATUS
FROM USER_INDEXES
WHERE STATUS <> 'VALID'
ORDER BY 1;
```

8.6. Change data owner schema storage

If tables and indexes are in the same tablespace then move them to separate tablespaces to reduce datafile fragmentation and to reduce datafile contention and to improve performance, this also improves the performance of the rebuild indexes maintenance task.

```
--GIS DATA OWNER USER
sqlplus /nolog
connect gis/****@aeropro1.esri.com
show con_name

--move tables
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS_DATA;' AS SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = 'GIS_INDEX'
UNION
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS_DATA;' AS SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = '' OR TABLESPACE_NAME IS NULL
ORDER BY 1;

--move indexes
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE GIS_INDEX;' AS
SQLMOVE
  FROM USER_INDEXES
    WHERE TABLESPACE_NAME = 'GIS_DATA' AND INDEX_TYPE = 'NORMAL' AND TEMPORARY = 'N'
UNION
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE GIS_INDEX;' AS
SQLMOVE
  FROM USER_INDEXES
    WHERE (TABLESPACE_NAME = '' OR TABLESPACE_NAME IS NULL) AND INDEX_TYPE = 'NORMAL' AND
TEMPORARY = 'N'
ORDER BY 1;
```

```
--GIS2 DATA OWNER USER
sqlplus /nolog
connect gis2/****@aeropro1.esri.com
show con_name

--move tables
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS2_DATA;' AS SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = 'GIS2_INDEX'
UNION
SELECT 'ALTER TABLE ' || TABLE_NAME || ' MOVE TABLESPACE GIS2_DATA;' AS SQLMOVE
  FROM USER_TABLES
    WHERE TABLESPACE_NAME = " OR TABLESPACE_NAME IS NULL
ORDER BY 1;

--move indexes
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE GIS2_INDEX;' AS
SQLMOVE
  FROM USER_INDEXES
    WHERE TABLESPACE_NAME = 'GIS2_DATA' AND INDEX_TYPE = 'NORMAL' AND TEMPORARY = 'N'
UNION
SELECT 'ALTER INDEX ' || INDEX_NAME || ' REBUILD ONLINE PARALLEL 4 TABLESPACE GIS2_INDEX;' AS
SQLMOVE
  FROM USER_INDEXES
    WHERE (TABLESPACE_NAME = " OR TABLESPACE_NAME IS NULL) AND INDEX_TYPE = 'NORMAL' AND
TEMPORARY = 'N'
ORDER BY 1;
```

```
--MOS: ALTER INDEX REBUILD ONLINE Fails with ORA-01450 (Doc ID 236329.1)
--Solution: do not use the "online" clause to rebuild the index
```

Then, verify invalid objects and invalid indexes again.

Note: read "7.2 Geodatabase SDE dbtune".

9. Prepare Geodatabase for Editing

9.1. Recreate the Editor and Viewer Users

```
/*------------------------------------------------------------------------
--GIS EDITOR USER
------------------------------------------------------------------------*/
DROP USER GIS_EDITOR CASCADE;
CREATE USER GIS_EDITOR PROFILE "GIS_USERS"
  IDENTIFIED BY *****
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
ALTER USER "GIS_EDITOR" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";
ALTER USER "GIS_EDITOR" PROFILE "GIS_USERS";
GRANT "CONNECT" TO "GIS_EDITOR";
GRANT "GIS_LOGIN" TO "GIS_EDITOR";
GRANT "GISEDITOR" TO "GIS_EDITOR";

/*------------------------------------------------------------------------
--GIS VIEWER USER
------------------------------------------------------------------------*/
DROP USER GIS_VIEWER CASCADE;
CREATE USER GIS_VIEWER PROFILE "GIS_USERS"
  IDENTIFIED BY *****
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
ALTER USER "GIS_VIEWER" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";
ALTER USER "GIS_VIEWER" PROFILE "GIS_USERS";
GRANT "CONNECT" TO "GIS_VIEWER";
GRANT "GIS_LOGIN" TO "GIS_VIEWER";
GRANT "GISVIEWER" TO "GIS_VIEWER";
```

```
/*-------------------------------------------------------------------
--GIS2 EDITOR USER
-------------------------------------------------------------------*/
DROP USER GIS2_EDITOR CASCADE;
CREATE USER GIS2_EDITOR PROFILE "GIS_USERS"
  IDENTIFIED BY *****
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
ALTER USER "GIS2_EDITOR" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";
ALTER USER "GIS2_EDITOR" PROFILE "GIS_USERS";
GRANT "CONNECT" TO "GIS2_EDITOR";
GRANT "GIS_LOGIN" TO "GIS2_EDITOR";
GRANT "GIS2EDITOR" TO "GIS2_EDITOR";

/*-------------------------------------------------------------------
--GIS2 VIEWER USER
-------------------------------------------------------------------*/
DROP USER GIS2_VIEWER CASCADE;
CREATE USER GIS2_VIEWER PROFILE "GIS_USERS"
  IDENTIFIED BY *****
  DEFAULT TABLESPACE "USERS"
  TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
ALTER USER "GIS2_VIEWER" TEMPORARY TABLESPACE "GIS_TEMP_GROUP";
ALTER USER "GIS2_VIEWER" PROFILE "GIS_USERS";
GRANT "CONNECT" TO "GIS2_VIEWER";
GRANT "GIS_LOGIN" TO "GIS2_VIEWER";
GRANT "GIS2VIEWER" TO "GIS2_VIEWER";
```

9.2. SDE Logfile Tables

```
--List sde logfile tables
SELECT *
FROM ALL_TABLES
WHERE TABLE_NAME LIKE 'SDE_LOG%' OR TABLE_NAME LIKE 'KEY%'
ORDER BY OWNER, TABLE_NAME;

--Temporary Tables
SELECT OWNER,TABLE_NAME, TABLESPACE_NAME, STATUS, TEMPORARY
FROM ALL_TABLES
WHERE (TABLE_NAME LIKE 'SDE_LOG%' OR TABLE_NAME LIKE 'KEY%' AND
TEMPORARY='Y'
ORDER BY 1,2;
```

```
--------------------------------------------------------------------------------
--Create sde logfile tables as global temporary tables
--------------------------------------------------------------------------------
```

sde_logfile_tables_gtt.sql

```
DROP TABLE sde_logfiles CASCADE CONSTRAINTS;
DROP TABLE sde_logfile_data CASCADE CONSTRAINTS;

CREATE GLOBAL TEMPORARY TABLE sde_logfiles
  (
   logfile_name VARCHAR2(256) NOT NULL ENABLE,
   logfile_id NUMBER(*,0) NOT NULL ENABLE,
   logfile_data_id NUMBER(*,0) NOT NULL ENABLE,
   registration_id NUMBER(*,0) NOT NULL ENABLE,
   flags NUMBER(*,0) NOT NULL ENABLE,
   session_tag NUMBER(*,0) NOT NULL ENABLE,
   logfile_data_db VARCHAR2(32),
   logfile_data_owner VARCHAR2(32),
   logfile_data_table VARCHAR2(98),
   column_name NVARCHAR2(32)
  )
ON COMMIT PRESERVE ROWS;

CREATE UNIQUE INDEX sde_logfiles_pk ON sde_logfiles (logfile_id);
CREATE UNIQUE INDEX sde_logfils_uk ON sde_logfiles(logfile_name);
CREATE UNIQUE INDEX sde_logfiles_uk2 ON sde_logfiles (logfile_data_id);

CREATE GLOBAL TEMPORARY TABLE sde_logfile_data
  (
   logfile_data_id NUMBER(*,0) NOT NULL ENABLE,
   sde_row_id NUMBER(*,0) NOT NULL ENABLE
  )
ON COMMIT PRESERVE ROWS;

CREATE INDEX sde_logfile_data_idx1 ON sde_logfile_data (logfile_data_id, sde_row_id);
CREATE INDEX sde_logfile_data_idx2 ON sde_logfile_data (sde_row_id);

DROP SEQUENCE SDE_LOGFILE_LID_GEN;

CREATE SEQUENCE SDE_LOGFILE_LID_GEN INCREMENT BY 1 START WITH 1 MAXVALUE 1.0E27 MINVALUE 1
NOCYCLE  CACHE 20 NOORDER;
```

```
-- Connect as each data owner user and each editor and viewer user and execute the sql script

sqlplus /nolog

connect gis/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

connect gis_editor/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

connect gis_viewer/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

connect gis2/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

connect gis2_editor/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

connect gis2_viewer/"*****"@mcsdboralnx7.esri.com:1577/aeropro1.esri.com
show con_name
@sde_logfile_tables_gtt.sql
disconnect;

--revoke gis_login role only from the editor and viewer users
revoke gis_login from gis_editor;
revoke gis_login from gis_viewer;
revoke gis_login from gis2_editor;
revoke gis_login from gis2_viewer;

exit;
```

For more information, see

- Log file tables in geodatabases in Oracle
- How To: Create the ArcSDE log file tables as global temporary tables in Oracle

Legacy:

For more information, see Log file table configuration options for geodatabases in Oracle.

9.3. Refresh Roles data owner users

createroles_gis.sql

```
----------------------------------------------------------------------------------
-- Developed: ESRI - Environmental Systems Research Institute Inc.
-- Objective: Create GIS roles
-- Historic:
-- Nov/04/2019 - Marcelo Marques - creation
----------------------------------------------------------------------------------
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;
SET SERVEROUTPUT ON;
spool Roles_gis.sql;

--DROP ROLE "GISEDITOR";
--CREATE ROLE "GISEDITOR" NOT IDENTIFIED;
--DROP ROLE "GISVIEWER";
--CREATE ROLE "GISVIEWER" NOT IDENTIFIED;

select 'grant select on ' ||owner|| '.' || table_name || '  to GISVIEWER;'
from sys.dba_tables
where lower(owner) = 'gis'
order by table_name;

select 'grant select,insert,update,delete on ' ||owner|| '.' || table_name || '  to GISEDITOR;'
from sys.dba_tables
where lower(owner) = 'gis'
order by table_name;

spool off;
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;

SET SERVEROUTPUT ON;
/
@Roles_gis.sql;
/
GRANT "GISEDITOR" TO "GIS_EDITOR";
GRANT "GISVIEWER" TO "GIS_VIEWER";
```

createroles_gis2.sql

```
---------------------------------------------------------------------------
-- Developed: ESRI - Environmental Systems Research Institute Inc.
-- Objective: Create GIS2 roles
-- Historic:
-- Nov/04/2019 - Marcelo Marques - creation
---------------------------------------------------------------------------
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;
SET SERVEROUTPUT ON;
spool Roles_gis2.sql;

--DROP ROLE "GIS2EDITOR";
--CREATE ROLE "GIS2EDITOR" NOT IDENTIFIED;
--DROP ROLE "GIS2VIEWER";
--CREATE ROLE "GIS2VIEWER" NOT IDENTIFIED;

select 'grant select on ' ||owner|| '.' || table_name || '  to GIS2VIEWER;'
from sys.dba_tables
where lower(owner) = 'gis2'
order by table_name;

select 'grant select,insert,update,delete on ' ||owner|| '.' || table_name || '  to GIS2EDITOR;'
from sys.dba_tables
where lower(owner) = 'gis2'
order by table_name;

spool off;
set echo off;
set verify off;
set heading off;
set feedback off;
set newpage none;
set termout off;
set lines 200;
set trims on;
ttitle off;
btitle off;
clear;

SET SERVEROUTPUT ON;
/
@Roles_gis2.sql;
/
GRANT "GIS2EDITOR" TO "GIS2_EDITOR";
GRANT "GIS2VIEWER" TO "GIS2_VIEWER";
```

9.4. Update Statistics

```
CREATE OR REPLACE PROCEDURE PROC_STATS AS
BEGIN
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName           => 'SDE'
    ,Granularity       => 'ALL'
    ,Options           => 'GATHER'
    ,Gather_Temp       => FALSE
    ,Degree            => 4
    ,Cascade           => TRUE
    ,No_Invalidate     => FALSE);
END;
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName           => 'GIS'
    ,Granularity       => 'ALL'
    ,Options           => 'GATHER'
    ,Gather_Temp       => FALSE
    ,Degree            => 8
    ,Cascade           => TRUE
    ,No_Invalidate     => FALSE);
END;
BEGIN
  SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
     OwnName           => 'GIS2'
    ,Granularity       => 'ALL'
    ,Options           => 'GATHER'
    ,Gather_Temp       => FALSE
    ,Degree            => 8
    ,Cascade           => TRUE
    ,No_Invalidate     => FALSE);
END;
END;
/

--execute store procedure
BEGIN
  SYS.PROC_STATS;
END;
/
```

9.5. Rebuild Indexes

Rebuild all indexes, attribute indexes and spatial indexes, then gather statistics again.

```
--execute store procedure
BEGIN
  SYS.PROC_REBUILD_INDEXES;
END;
/
```

See below the store procedure code.

```
CREATE OR REPLACE PROCEDURE PROC_REBUILD_INDEXES AS
    v_code  NUMBER;
    v_errm  VARCHAR2(64);
    CURSOR owner_cur IS
        SELECT DISTINCT(OWNER) OWNER from SDE.LAYERS ORDER BY OWNER;

    CURSOR index_cur IS
        SELECT owner, index_name FROM all_indexes
        WHERE owner in (select distinct(owner) from sde.layers) and INDEX_TYPE = 'NORMAL' AND TEMPORARY = 'N'
        ORDER BY owner, index_name;

    SQL_STMT VARCHAR2(200);

BEGIN
    DBMS_OUTPUT.ENABLE (buffer_size => NULL);
    FOR IndexRec in index_cur LOOP
        SQL_STMT := 'alter index ' || IndexRec.owner || '.' || IndexRec.index_name || ' rebuild online parallel 4';
        DBMS_OUTPUT.PUT_LINE('execute immediate: ' || SQL_STMT);
        BEGIN
          EXECUTE IMMEDIATE SQL_STMT;
        EXCEPTION
          WHEN OTHERS THEN
            v_code := SQLCODE;
            v_errm := SUBSTR(SQLERRM, 1, 64);
            DBMS_OUTPUT.ENABLE (buffer_size => NULL);
            DBMS_OUTPUT.PUT_LINE ('Error code ' || v_code || ': ' || v_errm);
            IF SQLCODE = -1450 THEN --MOS: ALTER INDEX REBUILD ONLINE Fails with ORA-01450 (Doc ID 236329.1)
              SQL_STMT := 'alter index ' || IndexRec.owner || '.' || IndexRec.index_name || ' rebuild parallel 4';
              DBMS_OUTPUT.PUT_LINE('execute immediate: ' || SQL_STMT);
              EXECUTE IMMEDIATE SQL_STMT;
           END IF;
        END;
    END LOOP;

    FOR OwnerRec in owner_cur LOOP
        --dbms_utility.analyze_schema (OwnerRec.owner, 'DELETE');
        --dbms_utility.analyze_schema (OwnerRec.owner, 'COMPUTE');
        BEGIN
          SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
                        OwnName           => OwnerRec.owner
            ,Granularity       => 'DEFAULT'
            ,Options           => 'GATHER'
            ,Gather_Temp       => FALSE
            ,Degree            => 8
            ,Cascade           => TRUE
            ,No_Invalidate     => FALSE);
        END;
    END LOOP;
    --dbms_utility.analyze_schema ('SDE', 'DELETE');
    --dbms_utility.analyze_schema ('SDE', 'COMPUTE');
    BEGIN
      SYS.DBMS_STATS.GATHER_SCHEMA_STATS (
        OwnName           => 'SDE'
        ,Granularity       => 'DEFAULT'
        ,Options           => 'GATHER'
        ,Gather_Temp       => FALSE
        ,Degree            => 4
        ,Cascade           => TRUE
        ,No_Invalidate     => FALSE);
    END;
EXCEPTION
  WHEN OTHERS THEN
  v_code := SQLCODE;
  v_errm := SUBSTR(SQLERRM, 1, 64);
  DBMS_OUTPUT.ENABLE (buffer_size => NULL);
  DBMS_OUTPUT.PUT_LINE ('Error code ' || v_code || ': ' || v_errm);
END;
```

9.6. Check Invalid Indexes

```
alter session set container = aeropro1
show con_name
select * from all_indexes d where d.status not in ('VALID','N/A');
```

9.7. Tablespaces Logging

```
/*
Disable Tablespace logging

Yes - Generate redo logs for creation of tables, indexes and partitions, and for subsequent inserts. Recoverable

No - Redo log entries are smaller; the above operations are not logged and not recoverable.

*/

sqlplus /nolog
connect / as sysdba;

SELECT owner , table_name
FROM dba_tables
WHERE logging='NO';

SELECT owner , table_name, index_name
FROM dba_indexes
WHERE logging='NO';

SELECT tablespace_name
FROM dba_tablespaces
WHERE logging='NOLOGGING';

SELECT TABLE_NAME, PARTITION_NAME
FROM user_tab_partitions
WHERE logging='NO';

SELECT 'ALTER TABLESPACE ' || CHR(34) || NAME || CHR(34) || ' LOGGING;'
  FROM V$TABLESPACE
  WHERE NAME NOT IN
  ('SYSTEM','SYSAUX','USERS','TEMP','TEMP2','TEMP3','TEMP4','UNDOTBS1')
ORDER BY NAME;

ALTER TABLESPACE "GIS_SDE" LOGGING;
ALTER TABLESPACE "GIS_SDEINDEX" LOGGING;
ALTER TABLESPACE "GIS_DATA" LOGGING;
ALTER TABLESPACE "GIS_INDEX" LOGGING;
ALTER TABLESPACE "GIS2_DATA" LOGGING;
ALTER TABLESPACE "GIS2_INDEX" LOGGING;
```

9.8. Pluggable Database Disable Force No Logging

If you have disabled “archivelog” then reenable “archivelog” and “flashback” and disable PDB “force no logging”.

a. Enable Archivelog

```
--Turn on Archive

--Windows
set oracle_sid=mcs

--Linux
.oraenv
mcs

------------------------------
sqlplus /nolog
connect /as sysdba
------------------------------
shutdown immediate;
startup mount;
alter database archivelog;
archive log list;
alter database flashback on;
select flashback_on from v$database;
alter database open;
select host_name,instance_name,status,database_status from v$instance;
select name,open_mode,log_mode,flashback_on from v$database;
```

b. Disable PDB Force No Logging

```
sqlplus /nolog
connect sys /as sysdba
----------------------------------------------------------------
--disaable pluggable database force nologging
ALTER SESSION SET CONTAINER = cdb$root;
show con_name
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE aeropro1 OPEN RESTRICTED;
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 DISABLE FORCE NOLOGGING;
ALTER PLUGGABLE DATABASE aeropro1 LOGGING;
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE;
ALTER PLUGGABLE DATABASE aeropro1 OPEN READ WRITE;
SELECT * FROM dba_pdbs ORDER BY pdb_name;
SELECT * FROM v$pdbs ORDER BY name;
----------------------------------------------------------------
--RAC
--disable pluggable database force nologging
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE INSTANCES=ALL;
--open restrictted pdb only on rac node 1
ALTER PLUGGABLE DATABASE aeropro1 OPEN RESTRICTED INSTANCES = ('mcs11');
ALTER SESSION SET CONTAINER = aeropro1;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 DISABLE FORCE NOLOGGING;
ALTER PLUGGABLE DATABASE aeropro1 LOGGING;
ALTER SESSION SET CONTAINER = cdb$root;
SHOW CON_NAME
ALTER PLUGGABLE DATABASE aeropro1 CLOSE IMMEDIATE INSTANCES=ALL;
ALTER PLUGGABLE DATABASE aeropro1 OPEN READ WRITE INSTANCES=ALL;
SELECT * FROM dba_pdbs ORDER BY pdb_name;
SELECT * FROM v$pdbs ORDER BY name;
----------------------------------------------------------------
```

9.9. RMAN Full Backup

Execute a rman full backup.

```
rman target sys/'*****'@mcs catalog rman_mcs/'*****'@rcatpdbmcs

--Incremental Level 0
RMAN> RUN {
backup incremental level 0 cumulative device type disk tag 'bkp_db_full' database;
backup device type disk tag 'bkp_archive' archivelog all not backed up delete all input;
delete noprompt obsolete device type disk;
}
```

community.esri.com

10.Esri Support Articles

FAQ: Which Oracle export/import utility should one use with ArcGIS and Oracle 10g?

https://support.esri.com/en/technical-article/000009842

FAQ: Working with ST_Geometry and Oracle export/import

https://support.esri.com/en/technical-article/000009852

How To: Perform schema restores of an Oracle enterprise geodatabase

https://support.esri.com/en/technical-article/000018063

FAQ: Are Oracle statistics for ST_Geometry attributes exported and imported?

https://support.esri.com/en/technical-article/000009858

How To: Detect detached st_spatial_index domain index tables in Oracle

https://support.esri.com/en/technical-article/000009845

Error: ORA-31684 'Index already exists' when importing ST_GEOMETRY spatial indexes using Oracle data pump utility (IMPDP).

https://support.esri.com/en/technical-article/000012233

11.Database Connections Best Practices

For more information, see community.esri.com.

12.Geodatabase Repository Upgrade

For more information, see Oracle eGDB SDE Repo Upgrade Using Oracle Restore Point

13.Database Template Scripts

How to install, configure, backup and maintain Very Large Enterprise MCS Databases (VLEMCSDB), detailed instructions for data loading, backup and recovery and to manage multiple user logins.

For Professional Database Administrators, requires advanced RDBMS and advanced Geodatabase experience.

The scripts cover the guide books best practices but also have very advanced RDBMS and advanced Geodatabase best practices for VLEMCSDB that go beyond the guide books recommendations.

For more information, see community.esri.com.

Note:

How Load Large Featureclass Oracle Geodatabase esri | THE SCIENCE OF WHERE®

Esri, the global market leader in geographic information system (GIS) software, offers the most powerful mapping and spatial analytics technology available.

Since 1969, Esri has helped customers unlock the full potential of data to improve operational and business results. Today, Esri software is deployed in more than 350,000 organizations including the world's largest cities, most national governments, 75 percent of Fortune 500 companies, and more than 7,000 colleges and universities. Esri engineers the most advanced solutions for digital transformation, the Internet of Things (IoT), and location analytics to inform the most authoritative maps in the world.

Visit us at esri.com.

Contact Esri

380 New York Street
Redlands, California 92373-8100 USA

1 800 447 9778
T 909 793 2853
F 909 793 5953
info@esri.com
esri.com

Offices worldwide
esri.com/locations