Building Databases after DBCA Creation on Azure Servers

# Abstract

This document describes the processes that should be followed in order to take a default database, created by DBCA, and make it resemble a production, pre-production, staging or test database for UV etc.

It is assumed that the database was created by following the instructions in the document *AZURE - Using DBCA to create a new database.docx* which can be found in *$TFS\TA\DEV\Projects\Oracle Upgrade 9i to 11g\UKRegulated\Database\DBA Documentation\Build Documents*. That process builds a standard blank general purpose database with the minimum of tablespaces etc.

In order to create a database ready to use in a UV manner, some additional build instructions must be followed to create the required tablespaces etc. These are detailed below.

# Initial Database Build

The database was initially built as a general purpose database using the DBCA.bat utility – as mentioned above. It was *not* configured for Enterprise Manager and was initially created in NOARCHIVELOG mode.

# Continuing the Database Build

To create a UV database we need some additional build processes. There are scripts available in TFS, at *$TFS\TA\DEV\Projects\Oracle Upgrade 9i to 11g\UKRegulated\Database\DBA Documentation\Code\TrialRunStuff\CFG\_BuildScripts*, which can be used to bring a default database up to UV requirements.

## Check T005\_Parameters

This is a script found in the above location, which sets numerous parameters to those required by an 11g database for UV use. The script will need editing on a case by case basis as not all databases will be required to be production ready, some are merely staging databases for various refreshes etc.

### Delete Obsolete Parameters

Check for, and remove if found, any and all of the following depreciated parameters from the script. The database will not start if these are present.

As of October 2016 this should not be required as the file in TFS has been correctly amended to remove the above – however, it pays to sanity check, just in case!

AQ\_TM\_PROCESSES (Cannot be explicitly set to zero on 11g).

CURSOR\_SPACE\_FOR\_TIME

FAST\_START\_IO\_TARGET

LOG\_ARCHIVE\_START

MAX\_ENABLED\_ROLES

PARALLEL\_AUTOMATIC\_TUNING

PARALLEL\_SERVER

PARALLEL\_SERVER\_INSTANCES

PLSQL\_V2\_COMPATIBILITY

REMOTE\_OS\_AUTHENT

SERIAL\_REUSE

SQL\_TRACE

DB\_BLOCK\_BUFFERS (Cannot be set, even to its default value, if SGA\_TARGET or MEMORY\_TARGET are present.)

### Adjust Database Usage Specific Parameters

Some parameters in the script are currently configured for production usage. This may not be appropriate, so ensure that the following are set according to the proposed usage of the database.

For production type usage, and test databases which are running in archivelog mode, including pre-production:

alter system set sga\_target=5g scope=spfile;

alter system set sga\_max\_size=6g scope=spfile;

alter system set pga\_aggregate\_target = 1g scope = spfile;

alter system set db\_recovery\_file\_dest\_size=500g scope=spfile;

Non production type databases, such as testing or staging, require lesser settings. The following are advised:

alter system set sga\_target=2g scope=spfile;

alter system set sga\_max\_size=3g scope=spfile;

alter system set pga\_aggregate\_target = 500m scope = spfile;

alter system set db\_recovery\_file\_dest\_size=100g scope=spfile;

Non-production databases usually run in NOARCHIVELOG mode, so do not need a huge FRA. If the database is to be run in ARCHIVELOG mode, adjust accordingly as per production.

### Adjust Standard Parameters

The following must be set as below regardless of the database usage.

alter system set streams\_pool\_size=300m scope=spfile;

alter system set log\_archive\_dest\_1='location=use\_db\_recovery\_file\_dest' scope=spfile;

alter system set log\_archive\_format='%D\_%S\_%R.%T.arc'scope=spfile;

### Always Turn Off Extra Cost Options

Oracle sneakily sets a couple of parameters to enable some extra cost options when using Enterprise Edition. We don’t pay for these options, so the must be disabled to avoid being fined in the event of a licencing audit.

alter system set control\_management\_pack\_access='NONE' scope=spfile;

alter system set enable\_ddl\_logging=false scope=both;

### Reset the Parameters

The parameters need to be applied to the database.

Login to the database as the SYSDBA user and execute the script. This will set the parameters in the spfile, only, ready to be applied at the next database start up.

set ORACLE\_HOME=C:\OracleDatabase\product\11.2.0\dbhome\_1

set ORACLE\_SID=CFG

set NLS\_DATE\_FORMAT=yyyy/mm/dd hh24:mi:ss

sqlplus sys/<password> as sysdba

@t005\_parameters.sql

Check the output from the above script before continuing. Fix any errors noted.

### Fixing SPFILE Errors

If you get a parameter setting wrong when setting it in the SPFILE, you will not know until the next start up. Oracle seems to validate settings at "memory" time, not at "spfile" time. The database will fail to start in this case.

This is easily fixed by creating a PFILE (initSID.ora) and setting the spfile parameter within to the existing spfile name, which has the broken parameter(s), and then, adding the corrections. For example, in REMOTE\_LISTENER has proven to prevent the database starting:

spfile='C:\OracleDatabase\product\11.2.0\dbhome\_1\spfileSID.ora'

REMOTE\_LISTENER = 'Correct Value'

Start the database in MOUNT mode using the above PFILE.

startup mount pfile='?\database\initSID.ora'

Then, for each incorrect parameter that you added to the PFILE, runSQL commands to correct them in the SPFILE for future use. For our example of REMOTE\_LISTENER, the following will suffice:

alter system set remote\_listener = 'Correct Value'

scope=spfile;

Once you have corrected everything in the spfile, startup the database with the corrections applied:

startup force

## Check and Edit T030\_Create\_redo\_log\_groups

All our databases end up with a set of redo logfiles, in groups of two members, which match the production databases. Each group should have a member on the oradata side, and a member on the fra.

The script creates log groups 4 through 13, inclusive, and adds two members, correctly placed, in each group. The two paths for the members are assumed to be:

* '??:\mnt\oradata\ORACLE\_SID\' for the oradata member; and
* '!!:\mnt\fast\_recovery\_area\ORACLE\_SID\' for the FRA member.

Once the new groups are created, the existing groups 1 through 3 are rotated out and dropped, leaving only the new groups present.

Obviously, you will have to edit the script to correctly identify the desired drives and database name in the two paths above.

* Edit t030\_create\_redo\_log\_groups.sql and:
  + Make sure that the various file and/or disc locations are correct. Replace '??:' with your desired oradata disc letter, and '!!:' with your desired FRA drive letter.
  + Replace '\ORACLE\_SID\' with the oracle database name.
  + Make sure that one of the redo log files in each group is on the FRA with the other in the oradata area.
  + Save the changes.
* Execute t030\_create\_redo\_log\_groups.sql:

@t030\_create\_redo\_log\_groups.sql

## Check and Edit T040\_Create\_tablespaces

This script creates any desired new tablespaces for a UV database. As with the redo logfile script, the drive letters have been obfuscated to prevent accidental execution with the wrong drive, or database name, specified.

* The script *T040\_create\_tablespaces.sql* was edited to fill in the details from the latest export reconciliation script for the Trial Run of the migration. This has all the commands required to create all the tablespaces, but requires a little editing to:
  + Make sure that the disc locations are correct. All data files are assumed to be created on the oradata path and are currently set to use '??:\mnt\oradata\ORACLE\_SID\' for the path.
    - Change '??:' to reflect your desired oradata path's drive letter.
    - Change '\ORACLE\_SID\' to reflect the correct database name.
  + Save the changes.

* Execute the script:

@t040\_create\_tablespaces.sql

It will take a while to complete as it is adding quite a few gigabytes of data files to the database.

## T060\_Create\_verify\_function

This script creates the default password verification function for the UV databases. It simply requires to be executed.

@t060\_create\_verify\_function.sql

## Post Creation Tasks

After the database has been built up to a UV standard, there is a little tidying up to carry out. Although the DBCA script was told not to create the demo schemas, it still creates a database with the scott schema present. This needs to be deleted for security purposes. All databases will have Statspack installed, but only production will actually utilise it (at present!)

set timing off

alter database force logging;

alter database enable block change tracking using file '!!:\mnt\fast\_recovery\_area\ORACLE\_SID\bct.dbf';

drop user scott cascade;

-- Install Statspack

@?\rdbms\admin\spcreate

-- You are now logged in to PERFSTAT, no longer SYS!

* Check all tablespaces as follows:

connect sys/<password> as sysdba

col gb format 9,990.99

set lines 2000 pages 2000 trimspool on

select tablespace\_name, sum(bytes)/1024/1024/1024 as gb

from dba\_data\_files

where tablespace\_name not in ('SYSTEM','SYSAUX','UNDOTBS1','XDB','DRSYS','TOOLS')

group by tablespace\_name

--

union all

--

select tablespace\_name, sum(bytes)/1024/1024/1024 as gb

from dba\_temp\_files

group by tablespace\_name

--

order by 1;

* The results should resemble the following:

TABLESPACE\_NAME GB

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

ARCHIVE1 0.49

ARCHIVE1\_INDEX 0.49

AURA 0.00

AURA\_INDEX 1.37

CFA 21.24

CFA\_INDEX 5.00

CFGLOG 4.65

CFGLOG\_INDEX 14.57

COA 3.61

COA\_INDEX 2.79

CWMLITE 0.02

FTREG 4.23

FTREG\_INDEX 2.02

INDX 0.02

ODM 0.02

PERFSTAT 2.00

SNAPLOG 0.10

TAKEON 3.56

TAKEON\_INDEX 1.12

TEMP 10.00

USERS 0.04

USERS\_INDEX 0.01

UVDATA01 1.00

UVDATA01\_INDEX 0.49

UVLOG01 1.00

UVLOG01\_INDEX 0.49

26 rows selected

* If the database is to be run in archivelog mode – currently only production and pre-production databases get this - then proceed as follows:

shutdown

startup mount

alter database archivelog;

alter database flashback on;

alter database open;

exit

* Update the tnsnames.ora file(s) on (at least) this Azure server, and, the others as required. Consider also updating the central tnsnames.ora file located in \\CFSLDSFP01\Apps.Net\Aura\ TNSNAMES\_CENTRE:

cd %ORACLE\_HOME%\network\admin

notepad tnsnames.ora

* Add a new entry for the newly created database.

This concludes initial configuration for the database. Running an import of data taken from either the 9i database currently (at the time of writing) or from another 11g Azure database, will create all the required grants etc.