Danny's Useful UDB and Oracle Commands

Note: These commands may not be used on a day to day basis. You may only ever need to run these whenever you encounter a specific issue. I have tried detailing the situation it would be required that out in each explanation.

UDB Commands

db2stop force:

This will kill the db2 instance and any process that might be running on it. ALWAYS USE THIS AS A LAST RESORT, it is extremely dangerous to kill a db2 instance when a process is still executing, and could leave tablespaces/tables in strange states, and the database in recovery mode. The preferred method is "**db2stop**" but this command won't work if something is running. You use the "force" option if you need to stop whatever is running (long running reorg, run away process, etc).

db2 list db directory
db2 list node directory

These two commands will list all the databases/nodes catalogued to that particular instance. Useful for when trying to figure out connection issues or when cataloging a new database to that instance, to make sure it is catalogued correctly.

db2level

This will last the path to the installation binaries and what version of DB2 the databases are running on. This is useful when doing uninstalls since you will need to location of the binaries to run the uninstall commands. Also useful when troubleshooting and you need to know what version of DB2 (each version has its own set of differences, so problem resolution may depend on the instance version).

*******DROPPING A UDB DATABASE AND INSTANCE*******

Before uninstall: run a final full database backup (usually located in the /instance/tools directory, with the following job name: <dbname>.udbdiskbu_hot and push the back up to TSM or Data Domain so that backup file is copied somewhere before it gets deleted off the file system)

1. **db2 drop db <database name> -** drops the database. Do this for all dbs under that instance
2. log in as **root** – You will need to root to run the uninstall command, open a ticket to AIX/Linux folks
3. cd to binaries folder (/opt/ibm/db2/version) (make sure to check where binaries are installed – that useful **db2level** command listed above)
4. cd to the instance folder
5. **db2idrop <inst name>** (make sure nothing is running by using **db2stop force** command)

. db2profile – rereads and sets everything that is listed in the db2 user profile. Sometimes it may not make your changes effective, if you have updated something in the db2 profile, so you have to log out of the instance and back in for them to take effect.

******LOOK UP WHERE YOU ARE IN A PROCESS (REORG, BACKUP, ETC)************

lutild (db2 list utilities show detail) – The first of many shortcuts Matt Soja created. This will list any utilities running on the database (backups and restores mostly). Useful for seeing how far along a particular backup or restore is, so that you can calculate how much longer it will take to complete.

db2pd -reorg -db -<db name> - This will show you how far along a table reorg is, and what step it is in (Sort, Build, IndxRct, Done). You can calculate how long it will take based on where it is in the process. If you want to see specifically (for large tables) how many indexes have been rebuilt and how many still remain, you can use the below command (be sure to be in the diaglog directory, usually /instance/db2dump

```
$ grep Rebuilding db2diag.log
MESSAGE : ADM5540W Rebuilding "9" indexes on table "WSADMIN .LSW_TASK".
MESSAGE : ADM5541W Rebuilding index with IID "1" in object with ID "12" and
MESSAGE : ADM5541W Rebuilding index with IID "2" in object with ID "12" and
MESSAGE : ADM5541W Rebuilding index with IID "3" in object with ID "12" and
MESSAGE : ADM5541W Rebuilding index with IID "4" in object with ID "12" and
```

touch reorg_tbls.term – This creates a “reorg_tbls.term” file (touch is the command to create an empty file in Unix/Linux). You would run this if a reorg has been long for a long time and you need to terminate it after the table it is reorg-ing completes. If you create this file in the tools directory, where the reorg job resides, it will kill the reorg before it starts on the next table.

DB2 COMM

db2set db2comm=tcpip
db2 update dbm cfg using SVCNAME db2c_instancename

These two commands are used when a new instance is created and you need to setup the databases so that outside users can connect to them. The “db2comm” parameter is for the specific database and the “SVCNAME” parameter is for the instance. You will also have to update the /etc/services file (requires **root** access) to include the “db2c_instancename” parm and the port number that will be used to connect to the instance/database.

```
db2 ? SQL5043N
SQL5043N  Support for one or more communications protocols failed to
      start successfully. However, core database manager functionality
      started successfully.
```

This error usually means something is wrong with either the SVCNAME or DB2COMM parameter.

----

db2look -d <dbname> -a -e -l -f -xd -o db2look.out – This creates a db2look file. This will list ALL of the information about that database, including all the tables, tablespaces, grants, etc. This is ESPECIALLY useful when running a restore of a database from production to a lower environment. When you restore the db into a lower environment it will lose all of the grants it has in place. However if you run this “db2look” command on the database in the lower environment, you will have all the original grants and you can run them after the database has been restored from the production copy.

db2 reorg table <table name> index <index name> use <tempspace>
db2 runstats on table <table name> and detailed indexes all shrlevel change

These are basic reorg/runstats commands. Useful to know when you want to reorg or runstat a certain table (we see a lot of issues with access paths that get resolved with runstats).

************REDIRECTED RESTORES***************

db2 restore db <dbname> from <backup directory> taken at <backup time> redirect generate script <script name>

The above command will generate a redirected restore script from the backup of the database you are looking to restore from. The script will list all the database containers. PLEASE NOTE that the file systems listed in the redirect script MAY not match up with the file systems on the server you are restoring TO. In that case you need to adjust the script accordingly before running it (you will get errors if it cannot find the correct file system to build the containers).

db2set DB2_RESTORE_GRANT_ADMIN_AUTHORITIES=YES

This database parameter will carry over the default authorities from the database that is being restored into. If you do not set this parameter PRIOR to running the restore the target environment's instance id will not have any authority, and thus you will not be able to grant any access to users with the instance ID. The only work around is to create the instance ID from the database that it was restored from and then grant access to the target instance ID (which is a hassle and not worth the effort).

----

db2 rollforward db <dbname> to end of backup and complete – Use the rollforward command after restoring a database from an online backup. Until you rollfoward the database the db will stay in "ROLL FORWARD PENDING" and thus inaccessible.

db2 alter tablespace <tablespace name> switch online – Use this command if the database has suffered a crash and the tablespace is in "Offline" status (you can check by using a 'db2 list tablespaces show detail').

ORACLE Commands

du –sh – A UNIX command that stands for disk usage. "h" will list the output in human readable format (listed in bytes B, K, M, G, T) and "s" will report only the sum of the usage in the current directory, not for each file. Useful for when to diagnose what directory has the most data in it and you need to clean it out (mostly for file system alerts).

****DATA FILES****

Select name, total_mb/1024 TOTAL_GB, free_mb/1024 FREE_GB from v$asm_diskgroup;

VERY useful SQL query to see how much space has been taken from a file system belonging to a data file. Use this to figure if you can add space to a particular data file from that file system.

Adding a data file via SQLplus:

```
ALTER TABLESPACE <tablespace name>
ADD DATAFILE '/tivwprd_01/tivwprd_itim_tblsp29.dbf'
SIZE 32666M
AUTOEXTEND OFF;
```

****Cleaning up /orabase****

In the alert directory (/orabase/diag/rdbms/<instance name>/<instance name>/alert), you can run the following command:

rm log_?.xml log_??.xml log_???.xml

This will remove all the log file except the one currently running (you DO NOT want to remove this one).

*****START AND STOP AN ORACLE DATABASE*****

srvctl status database -d <dbname> : checks whether the db is up and running
srvctl start database -d <dbname> : starts up a database (non rac)
srvctl stop database -d <dbname> : stops a database (non rac)

srvctl status/start/stop -i <instance name> : for RAC dbs (eg. ccdrprod1)

IF THE LISTENER IS DOWN (BUT DB IS RUNNING)

lsnrctl start listener_<dbname>

To select all of the tables from a certain schema (useful when you need to create a read-only/read-write role and need all of the tables in certain schema):

select * from dba_objects where owner='<schema name>' and object_type='TABLE'

To get the hashed value of a pw and reset a user using their existing pw:

select name, password from sys.user$ where name='XXXXXX';

Alter USER XXXX identified by values '<HASH VALUE>';

!stty erase ^H - so you can use backspace when running commands via sqlplus