CL SCR

/\* 1/30/2015 Andrew Bloss

Comment out the backups you do not want to upgrade.

Connect to SYS on the server with the backups to be tested.

Some schemas have been customized and require scripts to be run on them before the upgrade script.

Name these scripts <SCHEMA\_NAME\_IN\_BACKUP>.sql and put them in C:\BRM\_ORACLE\_UPGRADES\BEFORE\_UPGRADE.

1. Copy the upgrade script and paste it into C:\BRM\_ORACLE\_UPGRADES on the Oracle server.

2. Log on to sys on the server with the upgrades.

3. Run this script.

3. Copy the output after ------ and paste it and the script defined in UPGRADE\_SCRIPT into C:\BRM\_ORACLE\_UPGRADES\Upgrades.sql on the Oracle server.

4. Execute C:\BRM\_ORACLE\_UPGRADES\Upgrades.bat on the Oracle server (On wazdevbsora02 this runs SQLPLUS SYS/Pontis51$@PONTIS2 AS SYSDBA @C:\BRM\_ORACLE\_UPGRADES\Upgrades.sql).

SET UP MOVING THIS TO A NEW ORACLE SERVER.

Copy C:\BRM\_ORACLE\_UPGRADES on wazdevbswora02 to the C drive of your Oracle server.

Remove the SID (@PONTIS2) from all of the code below if the backups should be restored to the default SID

or replace PONTIS2 with the correct SID.

Change the sys password (Pontis51$) below if necessary.

\*/

VAR SUFFIX VARCHAR2(30);

VAR UPGRADE\_SCRIPT VARCHAR2(30);

BEGIN

-- New schema names are made by appending the letters in :SUFFIX to the schema name in the backup.

SELECT 'DEV' INTO :SUFFIX FROM DUAL;

-- Enter the name of the test script.

SELECT 'OracleBRM522.sql' INTO :UPGRADE\_SCRIPT FROM DUAL;

END;

/

SET SERVEROUTPUT ON

-- Abort if any schemas to be upgraded have sessions connected to them.

WHENEVER SQLERROR EXIT SQL.SQLCODE ROLLBACK;

DECLARE

V\_X PLS\_INTEGER;

BEGIN

SELECT COUNT(\*) INTO V\_X FROM ALL\_TABLES WHERE TABLE\_NAME = 'PON\_BACKUPS';

IF (V\_X = 1) THEN

EXECUTE IMMEDIATE 'DROP TABLE PON\_BACKUPS';

END IF;

EXECUTE IMMEDIATE 'CREATE TABLE PON\_BACKUPS(

BACKUP\_NAME VARCHAR2(60),

SCHEMA\_NAME\_IN\_BACKUP VARCHAR2(30),

NEW\_SCHEMA\_NAME VARCHAR2(30),

BEFORE\_UPGRADE VARCHAR2(60))';

END;

/

INSERT INTO PON\_BACKUPS(BACKUP\_NAME, SCHEMA\_NAME\_IN\_BACKUP, NEW\_SCHEMA\_NAME, BEFORE\_UPGRADE)

SELECT 'ALASKA\_4X\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'ALASKA\_521\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'CALIFORNIA\_513\_ORIG', NULL, NULL, 'BEFORE\_CALIFORNIA\_513\_ORIG.SQL' FROM DUAL UNION

SELECT 'FLORIDA\_45\_ORIG', NULL, NULL, 'BEFORE\_FLORIDA\_45\_ORIG.SQL' FROM DUAL UNION

SELECT 'IDAHO\_4X\_ORIG', NULL, NULL, 'BEFORE\_IDAHO\_4X\_ORIG' FROM DUAL UNION

SELECT 'IDAHO\_513\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'IDAHO\_521\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'IOWA\_44\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'KANSAS\_443\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'KENTUCKY\_5103\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'KENTUCKY\_521\_SP2\_ORIG', NULL, NULL, 'BEFORE\_KENTUCKY\_521\_SP2\_ORIG.SQL' FROM DUAL UNION

SELECT 'LA\_COUNTY\_521\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'MAINE\_4X\_ORIG', NULL, NULL, 'BEFORE\_MAINE\_4X\_ORIG.SQL' FROM DUAL UNION

SELECT 'MINNESOTA\_44\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'MISSISSIPPI\_40\_ORIG', NULL, NULL, 'BEFORE\_MISSISSIPPI\_40\_ORIG.SQL' FROM DUAL UNION

SELECT 'MONTANA\_NP\_ORIG', NULL, NULL, 'BEFORE\_MONTANA\_NP\_ORIG.SQL' FROM DUAL UNION

SELECT 'NEW\_JERSEY\_43\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'TENNESSEE\_40\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'VIRGINIA\_5X\_ORIG', NULL, NULL, NULL FROM DUAL UNION

SELECT 'WISCONSIN\_513\_ORIG', NULL, NULL, NULL FROM DUAL;

UPDATE PON\_BACKUPS

SET SCHEMA\_NAME\_IN\_BACKUP = BACKUP\_NAME

WHERE SCHEMA\_NAME\_IN\_BACKUP IS NULL;

UPDATE PON\_BACKUPS

SET NEW\_SCHEMA\_NAME = SUBSTR(SCHEMA\_NAME\_IN\_BACKUP, 1, 30 - (LENGTH(:SUFFIX) + 1)) || '\_' || :SUFFIX

WHERE NEW\_SCHEMA\_NAME IS NULL;

SET SERVEROUTPUT ON

BEGIN

DBMS\_OUTPUT.ENABLE();

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('Close these connections.');

END;

/

SELECT 'CLOSE THE CONNECTION TO ' || B.NEW\_SCHEMA\_NAME || ' AND RERUN THIS SCRIPT.'

FROM PON\_BACKUPS B

JOIN V$SESSION VS ON VS.USERNAME = B.NEW\_SCHEMA\_NAME;

-- ABORT if there are connections to one of the schema to be upgraded. Schemas with connections can not be dropped.

DECLARE

V\_COUNT\_CONNECTED\_USERS PLS\_INTEGER;

BEGIN

SELECT COUNT(\*) INTO V\_COUNT\_CONNECTED\_USERS

FROM PON\_BACKUPS B

JOIN V$SESSION VS ON VS.USERNAME = B.NEW\_SCHEMA\_NAME;

IF (V\_COUNT\_CONNECTED\_USERS > 0) THEN

DBMS\_OUTPUT.PUT\_LINE('Disconnected sessions to the above users and rerun this script.');

RAISE\_APPLICATION\_ERROR(-20000, 'DISCONNECT USERS');

END IF;

END;

/

DECLARE

V\_Q VARCHAR2(300);

V\_X PLS\_INTEGER;

BEGIN

DBMS\_OUTPUT.ENABLE();

DBMS\_OUTPUT.PUT\_LINE('Copy everthing after this, paste it and ' || :UPGRADE\_SCRIPT || ' into C:\BRM\_ORACLE\_UPGRADES\Upgrades.sql and run Upgrades.bat.');

DBMS\_OUTPUT.PUT\_LINE('--------------------------------------------------------------------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('SPOOL C:\BRM\_ORACLE\_UPGRADES\' || :UPGRADE\_SCRIPT || '.out REPLACE');

DBMS\_OUTPUT.PUT\_LINE('SET LINESIZE 999');

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('');

SELECT COUNT(\*) INTO V\_X FROM PON\_BACKUPS;

DBMS\_OUTPUT.PUT\_LINE('-- Upgrade test script created on ' || TO\_CHAR(SYSDATE, 'MM/DD/YYYY') || ' for:');

FOR X IN (

SELECT NEW\_SCHEMA\_NAME

FROM PON\_BACKUPS

ORDER BY NEW\_SCHEMA\_NAME

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('-- ' || X.NEW\_SCHEMA\_NAME);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('-- ' || V\_X || ' upgrades.');

DBMS\_OUTPUT.PUT\_LINE('');

-- Print out statements to drop user, create new user, grant privileges, import, run before upgrade script and run upgrade on each backup.

FOR X IN (

SELECT \* -- B.BACKUP\_NAME, B.SCHEMA\_NAME\_IN\_BACKUP, B.NEW\_SCHEMA\_NAME, B.BEFORE\_UPGRADE, U.USERNAME

FROM PON\_BACKUPS B

LEFT OUTER JOIN ALL\_USERS U ON U.USERNAME = B.NEW\_SCHEMA\_NAME

ORDER BY B.NEW\_SCHEMA\_NAME

)

LOOP

IF (X.USERNAME IS NOT NULL) THEN

V\_Q := 'DROP USER ' || X.NEW\_SCHEMA\_NAME || ' CASCADE';

DBMS\_OUTPUT.PUT\_LINE(V\_Q);

DBMS\_OUTPUT.PUT\_LINE('/');

END IF;

V\_Q := 'CREATE USER ' || X.NEW\_SCHEMA\_NAME || ' IDENTIFIED BY pontis DEFAULT TABLESPACE USERS';

DBMS\_OUTPUT.PUT\_LINE(V\_Q);

DBMS\_OUTPUT.PUT\_LINE('/');

V\_Q := 'GRANT ALL PRIVILEGES TO ' || X.NEW\_SCHEMA\_NAME;

DBMS\_OUTPUT.PUT\_LINE(V\_Q);

DBMS\_OUTPUT.PUT\_LINE('/');

DBMS\_OUTPUT.PUT\_LINE('HOST IMPDP ' || X.NEW\_SCHEMA\_NAME ||'/pontis@PONTIS2 DIRECTORY=ORACLE\_BACKUPS DUMPFILE=' || X.BACKUP\_NAME || '.DMP REMAP\_SCHEMA=' || X.SCHEMA\_NAME\_IN\_BACKUP || ':' || X.NEW\_SCHEMA\_NAME || ' LOGFILE=' || X.BACKUP\_NAME || '\_IMP.LOG');

DBMS\_OUTPUT.PUT\_LINE('CONNECT ' || X.NEW\_SCHEMA\_NAME || '/pontis@PONTIS2');

IF (X.BEFORE\_UPGRADE IS NOT NULL) THEN

DBMS\_OUTPUT.PUT\_LINE('@C:\BRM\_ORACLE\_UPGRADES\BEFORE\_UPGRADE\' || X.BEFORE\_UPGRADE);

END IF;

IF (:UPGRADE\_SCRIPT IS NOT NULL) THEN

DBMS\_OUTPUT.PUT\_LINE('@C:\BRM\_ORACLE\_UPGRADES\' || :UPGRADE\_SCRIPT);

END IF;

DBMS\_OUTPUT.PUT\_LINE('');

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('EXIT');

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('-- Copy everything above this until -------------------------------------- ');

DBMS\_OUTPUT.PUT\_LINE('-- and paste it into C:\BRM\_ORACLE\_UPGRADES\UPGRADES.SQL on the server.');

DBMS\_OUTPUT.PUT\_LINE('-- Save it!');

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('-- Copy ' || :UPGRADE\_SCRIPT || ' and paste it into C:\BRM\_ORACLE\_UPGRADES\ on the server.');

DBMS\_OUTPUT.PUT\_LINE('');

DBMS\_OUTPUT.PUT\_LINE('-- Run C:\BRM\_ORACLE\_UPGRADES\Upgrades.bat on the server.');

END;

/