

Migrando una Base de Datos versión 10gR2 sobre HPUX hacia 11gR2 en AIX mediante Transportable Database

Por Francisco Riccio 

Introducción

El objetivo de este artículo es presentar un procedimiento validado de cómo realizar una migración de Base de Datos versión 10.2.0.4 sobre una plataforma HP-UX B.11.31 (ia64 hp server rx3600) hacia una versión 11.2.0.3 en plataforma IBM AIX 7.1 (Power 750).

Transportable Database es la técnica que se utilizará para la migración entre plataformas distintas, logrando un tiempo de indisponibilidad mucho menor a un export e import datapump.

Pasos:

Los pasos descritos en este artículo están divididos en dos fases, las cuales son:

Fase 1: Se realizará la migración de la base de datos entre plataformas distintas. Migraremos una base de datos versión 10.2.0.4 en HP-UX B.11.31 a versión 10.2.0.4 en IBM AIX 7.1.

El origen se encuentra en filesystems de tipo VXFS y el destino trabajará con ASM. El servidor destino cuenta con una instancia ASM versión 11.2.0.3. El diskgroup de ASM disponible para albergar la base de datos debe tener el atributo compatible.rdbms con un valor menor o igual a 10.2.

Como requisito en esta fase el servidor origen y destino deben trabajar inicialmente con la misma versión de base de datos.

Fase 2: Una vez migrado la base de datos a la plataforma AIX 7.1 se realizará el upgrade de base de datos de versión 10.2.0.4 a 11.2.0.3.

Como requisito debemos tener instalado el software de base de datos versión 11.2.0.3.

I. **Fase 1:**

A. **Base de Datos Origen:**

1. Validación de que ambos sistemas operativos tengan el mismo endian format.

```
SQL> select * from v$transportable_platform;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
7	Microsoft Windows IA (32-bit)	Little
10	Linux IA (32-bit)	Little
6	AIX-Based Systems (64-bit)	Big
3	HP-UX (64-bit)	Big
5	HP Tru64 UNIX	Little
4	HP-UX IA (64-bit)	Big
11	Linux IA (64-bit)	Little
15	HP Open VMS	Little
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
13	Linux x86 64-bit	Little
16	Apple Mac OS	Big
12	Microsoft Windows x86 64-bit	Little
17	Solaris Operating System (x86)	Little
18	IBM Power Based Linux	Big
19	HP IA Open VMS	Little
20	Solaris Operating System (x86-64)	Little
21	Apple Mac OS (x86-64)	Little

20 rows selected.

En nuestro caso HP-UX IA (64 bit) y AIX-Based Systems (64-bit) tiene el mismo valor de endian format por lo cual se puede proceder en la implementación.

2. Debemos validar si tenemos tablas externas, objetos directorios y BFILES en la base de datos.

```
set serveroutput on;  
  
declare x boolean;  
  
begin x := dbms_tdb.check_external;  
  
end;  
  
/
```

```

SQL> set serveroutput on;
      declare x boolean;
            begin x := dbms_tdb.check_external;
            end;
      /
SQL>                                     The following external tables exist in the database:
SH.SALES_TRANSACTIONS_EXT
The following directories exist in the database:
SYS.SUBDIR, SYS.XMLDIR, SYS.MEDIA_DIR, SYS.LOG_FILE_DIR, SYS.DATA_FILE_DIR,
SYS.WORK_DIR, SYS.ADMIN_DIR, SYS.ORACLE_OCM_CONFIG_DIR, SYS.DATA_PUMP_DIR
The following BFILES exist in the database:
PM.PRINT_MEDIA

PL/SQL procedure successfully completed.

```

Podemos apreciar que tenemos una serie de objetos listados, estos serán trabajados en el punto I.B.12.

3. Colocar la base de datos origen en modo READ ONLY

```

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.

Total System Global Area 167772160 bytes
Fixed Size                 2054672 bytes
Variable Size             100664816 bytes
Database Buffers          58720256 bytes
Redo Buffers               6332416 bytes
Database mounted.
SQL> alter database open read only;

Database altered.

```

4. El siguiente script tiene como finalidad validar que la base de datos cumpla con los requerimientos necesarios para migrar hacia la otra plataforma que deseamos. Cualquier error que se muestre debe ser corregido.

```

set serveroutput on

DECLARE

  db_ready BOOLEAN;

BEGIN

  db_ready := DBMS_TDB.CHECK_DB('AIX-Based Systems (64-bit)',

```

```
DBMS_TDB.SKIP_NONE);
```

```
END;
```

```
/
```

```
SQL> set serveroutput on
```

```
DECLARE
```

```
    db_ready BOOLEAN;
```

```
BEGIN
```

```
    db_ready := DBMS_TDB.CHECK_DB('AIX-Based Systems (64-bit)', DBMS_TDB.SKIP_NONE);
```

```
END;
```

```
/
```

```
SQL>
```

```
PL/SQL procedure successfully completed.
```

Como la salida del script no devuelve ningún mensaje de error procedemos a ejecutar los siguientes puntos.

5. Copiar los datafiles que se encuentran en el servidor origen hacia el servidor destino.

```
SQL> select file_name from dba_data_files;
```

```
FILE_NAME
```

```
-----  
/export/data/ORCL/users01.dbf  
/export/data/ORCL/sysaux01.dbf  
/export/data/ORCL/undotbs01.dbf  
/export/data/ORCL/system01.dbf  
/export/data/ORCL/example01.dbf
```

Mediante el comando SCP copiaremos los datafiles del servidor origen al destino.

```
$ scp /export/data/ORCL/*.dbf oracle@10.72.2.155:/u01/oradata/ORCL/.
```

```
oracle@10.72.2.155's password:
```

```
example01.dbf          100% 100MB   8.3MB/s   00:12  
sysaux01.dbf          100% 280MB  13.3MB/s   00:21  
system01.dbf         100% 490MB  11.1MB/s   00:44  
temp01.dbf           100%  20MB  20.0MB/s   00:01  
undotbs01.dbf        100%  25MB  25.0MB/s   00:01  
users01.dbf          100% 5128KB   5.0MB/s   00:01
```

6. Debemos identificar que datafiles contienen información de UNDO; donde serán los únicos que le aplicaremos una conversión para que sean compatibles con el nuevo sistema operativo.

```
select FILE_NAME from DBA_DATA_FILES
```

```
where TABLESPACE_NAME in
```

```
(select distinct TABLESPACE_NAME
from DBA_ROLLBACK_SEGS);
```

```
SQL> select FILE_NAME from DBA_DATA_FILES
where TABLESPACE_NAME in
(select distinct TABLESPACE_NAME
from DBA_ROLLBACK_SEGS);
```

FILE_NAME

```
-----
/export/data/ORCL/undotbs01.dbf
/export/data/ORCL/system01.dbf
```

Cabe mencionar que siempre aparecerán en la lista los datafiles pertenecientes al tablespace SYSTEM y de todos los tablespaces UNDO.

7. Copiaremos el PFILE de la base de origen al destino.

```
SQL> create pfile from spfile;
```

File created.

```
SQL> exit
```

```
$ scp $ORACLE_HOME/dbs/initORCL.ora oracle@10.72.2.155:/u01/oradata/ORCL/.
oracle@10.72.2.155's password:
initORCL.ora                                100% 1031    1.0KB/s   00:00
```

8. Creamos el script de recreación del control file y luego lo copiaremos al servidor destino.

```
SQL> alter database backup controlfile to trace resetlogs;
```

Database altered.

Se creará en archivo trace en nuestra carpeta de traces (USER_DUMP_DEST).

```
May 16 01:29 orcl_ora_17723.trc
```

Realizamos la copia del archivo trace al otro servidor.

```
$ scp orcl_ora_17723.trc oracle@10.72.2.155:/u01/oradata/ORCL/.
oracle@10.72.2.155's password:
orcl_ora_17723.trc                          100% 3690    3.6KB/s   00:00
```

B. Base de Datos Destino:

1. Modificar el PFILE con la finalidad que ellos sean válidos en el nuevo ambiente.

```
*.compatible='10.2.0.3.0'  
*.control_files='+DG_SEGA_DATA'  
*.db_block_size=8192  
*.db_name='ORCL'  
*.db_create_file_dest='+DG_SEGA_DATA'  
*.db_create_online_log_dest_1='+DG_SEGA_DATA'  
*.db_recovery_file_dest='+DG_SEGA_FRA'  
*.db_recovery_file_dest_size=2147483648  
*.dispatchers='(PROTOCOL=TCP) (SERVICE=ORCLXDB) '  
*.job_queue_processes=10  
*.open_cursors=300  
*.pga_aggregate_target=104857600  
*.processes=150  
*.remote_login_passwordfile='EXCLUSIVE'  
*.sga_target=267772160  
*.undo_management='AUTO'  
*.undo_tablespace='UNDOTBS1'
```

Por ejemplo en nuestro caso configuraremos las nuevas ubicaciones de los datafiles y del Flash Recovery Area en ASM. Asimismo habilitamos OMF.

2. Levantar en modo NOMOUNT con el PFILE copiado.

```
cp initORCL.ora $ORACLE_HOME/dbs/.  
export ORACLE_SID=ORCL
```

```
SQL> startup nomount  
ORACLE instance started.
```

```
Total System Global Area  267227136 bytes  
Fixed Size                  2220640 bytes  
Variable Size               130026912 bytes  
Database Buffers           130023424 bytes  
Redo Buffers                 4956160 bytes
```

3. Realizamos la conversión de los archivos identificados en la sección I.5 mediante el utilitario RMAN.

En esta migración estamos llevando los datafiles identificados hacia ASM ya con la conversión realizada.

```
run{  
  
  CONVERT DATAFILE '/u01/oradata/ORCL/system01.dbf'  
  
  FROM PLATFORM 'HP-UX IA (64-bit)'
```

```

FORMAT '+DG_SEGA_DATA';

CONVERT DATAFILE '/u01/oradata/ORCL/undotbs01.dbf'

FROM PLATFORM 'HP-UX IA (64-bit)'

FORMAT '+DG_SEGA_DATA';

}

```

```

RMAN> run{
CONVERT DATAFILE '/u01/oradata/ORCL/system01.dbf'
FROM PLATFORM 'HP-UX IA (64-bit)'
FORMAT '+DG_SEGA_DATA';
CONVERT DATAFILE '/u01/oradata/ORCL/undotbs01.dbf'
FROM PLATFORM 'HP-UX IA (64-bit)'
FORMAT '+DG_SEGA_DATA';
}

```

```

Starting conversion at target at 16-MAY-12
using target database control file instead of recovery catalog
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=226 device type=DISK
allocated channel: ORA_DISK_2
channel ORA_DISK_2: SID=242 device type=DISK
channel ORA_DISK_1: starting datafile conversion
input file name=/u01/oradata/ORCL/system01.dbf
channel ORA_DISK_2: starting datafile conversion
input file name=/u01/oradata/ORCL/undotbs01.dbf
converted datafile=+DG_SEGA_DATA/orcl/datafile/undotbs1.257.783392853
channel ORA_DISK_2: datafile conversion complete, elapsed time: 00:00:03
converted datafile=+DG_SEGA_DATA/orcl/datafile/system.258.783392853
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:07
Finished conversion at target at 16-MAY-12

```

4. Procedemos a copiar el resto de datafiles hacia ASM.

```

run{

CONVERT DATAFILE '/u01/oradata/ORCL/sysaux01.dbf'

FORMAT '+DG_SEGA_DATA';

CONVERT DATAFILE '/u01/oradata/ORCL/example01.dbf'

FORMAT '+DG_SEGA_DATA';

CONVERT DATAFILE '/u01/oradata/ORCL/users01.dbf'

```

```
FORMAT '+DG_SEGA_DATA';  
}
```

```
RMAN> run{  
  CONVERT DATAFILE '/u01/oradata/ORCL/sysaux01.dbf'  
  FORMAT '+DG_SEGA_DATA';  
  CONVERT DATAFILE '/u01/oradata/ORCL/example01.dbf'  
  FORMAT '+DG_SEGA_DATA';  
  CONVERT DATAFILE '/u01/oradata/ORCL/users01.dbf'  
  FORMAT '+DG_SEGA_DATA';  
}
```

```
Starting conversion at target at 16-MAY-12  
using target database control file instead of recovery catalog  
allocated channel: ORA_DISK_1  
channel ORA_DISK_1: SID=210 device type=DISK  
allocated channel: ORA_DISK_2  
channel ORA_DISK_2: SID=226 device type=DISK  
allocated channel: ORA_DISK_3  
channel ORA_DISK_3: SID=242 device type=DISK  
channel ORA_DISK_1: starting datafile conversion  
input file name=/u01/oradata/ORCL/sysaux01.dbf  
channel ORA_DISK_2: starting datafile conversion  
input file name=/u01/oradata/ORCL/example01.dbf  
channel ORA_DISK_3: starting datafile conversion  
input file name=/u01/oradata/ORCL/users01.dbf  
converted datafile=+DG_SEGA_DATA/orcl/datafile/sysaux.286.783393385  
channel ORA_DISK_1: datafile conversion complete, elapsed time: 00:00:07  
converted datafile=+DG_SEGA_DATA/orcl/datafile/example.287.783393385  
channel ORA_DISK_2: datafile conversion complete, elapsed time: 00:00:07  
converted datafile=+DG_SEGA_DATA/orcl/datafile/users.289.783393385  
channel ORA_DISK_3: datafile conversion complete, elapsed time: 00:00:07  
Finished conversion at target at 16-MAY-12
```

5. Recreación del control file.

Modificaremos el archivo trace del control file que creamos en el punto 1.7.

Para el siguiente caso se modificó la ubicación de cada datafile y redo log hacia las nuevas ubicaciones en ASM.

```
CREATE CONTROLFILE REUSE DATABASE "ORCL" RESETLOGS  
NOARCHIVELOG  
  
  MAXLOGFILES 16  
  
  MAXLOGMEMBERS 3
```



```
MAXDATAFILES 100

MAXINSTANCES 8

MAXLOGHISTORY 292

LOGFILE

GROUP 1 '+DG_SEGA_DATA/ORCL/redo01.log' SIZE 50M,
GROUP 2 '+DG_SEGA_DATA/ORCL/redo02.log' SIZE 50M,
GROUP 3 '+DG_SEGA_DATA/ORCL/redo03.log' SIZE 50M

-- STANDBY LOGFILE

DATAFILE

'+DG_SEGA_DATA/ORCL/system.258.783392853',
'+DG_SEGA_DATA/ORCL/undotbs1.257.257.783392853',
'+DG_SEGA_DATA/ORCL/sysaux.286.286.783393385',
'+DG_SEGA_DATA/ORCL/users.289.783393385',
'+DG_SEGA_DATA/ORCL/example.287.783393385'

CHARACTER SET WE8ISO8859P1

;
```

```

SQL> CREATE CONTROLFILE REUSE DATABASE "ORCL" RESETLOGS NOARCHIVELOG
      MAXLOGFILES 16
      MAXLOGMEMBERS 3
      MAXDATAFILES 100
      MAXINSTANCES 8
      MAXLOGHISTORY 292
LOGFILE
  GROUP 1 '+DG_SEGA_DATA/ORCL/redo01.log' SIZE 50M,
  GROUP 2 '+DG_SEGA_DATA/ORCL/redo02.log' SIZE 50M,
  GROUP 3 '+DG_SEGA_DATA/ORCL/redo03.log' SIZE 50M
-- STANDBY LOGFILE
DATAFILE
  '+DG_SEGA_DATA/ORCL/system.258.783392853',
  '+DG_SEGA_DATA/ORCL/undotbs1.257.257.783392853',
  '+DG_SEGA_DATA/ORCL/sysaux.286.286.783393385',
  '+DG_SEGA_DATA/ORCL/users.289.783393385',
  '+DG_SEGA_DATA/ORCL/example.287.783393385'
CHARACTER SET WE8ISO8859P1
;

```

Control file created.

6. Abrimos la base de datos en modo RESETLOGS.

```
SQL> alter database open resetlogs;
```

Database altered.

7. Creamos los tempfiles en ASM.

```
SQL> alter tablespace TEMP add tempfile '+DG_SEGA_DATA' size 1G;
```

Tablespace altered.

8. Luego procedemos a bajar la base de datos y la subiremos en modo upgrade.

```
SQL> shutdown immediate;
```

Database closed.

Database dismounted.

ORACLE instance shut down.

```
SQL> startup upgrade
```

ORACLE instance started.

```

Total System Global Area  771751936 bytes
Fixed Size                 2087192 bytes
Variable Size             201328360 bytes
Database Buffers         562036736 bytes
Redo Buffers              6299648 bytes
Database mounted.
Database opened.

```

9. Ejecutaremos el siguiente script: \$ORACLE_HOME/rdbms/admin/utlirp.sql

Nota: Si la base de datos tiene instalado el componente de Java debemos ejecutar el siguiente script antes de ejecutar el script utlirp.sql.

```
SQL> connect / as sysdba

SQL>

begin

update obj$ set status=5 where obj#=(select obj# from obj$,javasnm$
  where owner#=0 and type#=29 and short(+)=name and
  nvl(longdbcs,name)='oracle/aurora/rdbms/Compiler');

commit;

declare

cursor C1 is select
  'DROP JAVA DATA "' || u.name || "." || o.name || "'
  from obj$ o,user$ u where o.type#=56 and u.user#=o.owner#;
  ddl_statement varchar2(200);

iterations number;

previous_iterations number;

loop_count number;

my_err    number;

begin

previous_iterations := 10000000;

loop

  -- To make sure we eventually stop, pick a max number of iterations
  select count(*) into iterations from obj$ where type#=56;

  exit when iterations=0 or iterations >= previous_iterations;

  previous_iterations := iterations;
```

```

loop_count := 0;

open C1;

loop

begin

    fetch C1 into ddl_statement;

    exit when C1%NOTFOUND or loop_count > iterations;

exception when others then

    my_err := sqlcode;

    if my_err = -1555 then -- snapshot too old, re-execute fetch query

        exit;

    else

        raise;

    end if;

end;

initjvmaux.exec(ddl_statement);

loop_count := loop_count + 1;

end loop;

close C1;

end loop;

end;

commit;

initjvmaux.drp('delete from java$policy$shared$table');

update obj$ set status=1 where obj#=(select obj# from obj$,javasnm$

where owner#=0 and type#=29 and short(+)=name and

nvl(longdbcs,name)='oracle/aurora/rdbms/Compiler');

```

```

commit;

end;

/

create or replace java system

/

```

Si deseamos validar que tenemos instalado el componente Java en nuestra base de datos, ejecutamos el siguiente query el cual devolverá al menos una fila si está instalado.

```
SQL> select comp_name from dba_registry
      where comp_name like '%JAVA%';
```

```
COMP_NAME
```

```
-----
JServer JAVA Virtual Machine
```

10. Ejecutaremos el siguiente script: \$ORACLE_HOME/rdbms/admin/ catupgrd.sql.
11. Debemos reiniciar la base de datos en modo READ WRITE y ejecutamos el siguiente script: \$ORACLE_HOME/rdbms/admin/utlrp.sql
12. Actualización de los objetos identificados en el paso I.A.2

Por cada directorio y tabla externa que ha sido actualizado en el punto I.2 debe ser reemplazado por directorios válidos en el nuevo servidor, asimismo es necesario copiar el contenido de los directorios listados del servidor origen al destino.

Ejemplo:

Nuestro objeto directorio apunta a un directorio en el servidor origen que se muestra a continuación:

```
SQL> select directory_path from dba_directories where directory_name = 'DATA_PUMP_DIR';
```

```
DIRECTORY_PATH
```

```
-----
/u03/app/oracle/product/10.2.0/bd/rdbms/log/
```

Debemos crear el directorio en el servidor destino o en caso contrario realizar el siguiente comando:

```
CREATE OR REPLACE DIRECTORY DATA_PUMP_DIR AS '<nueva_ruta_dir>';
```

Luego procedemos a copiar el contenido del directorio original al nuevo si es que lo requerimos para nuestras aplicaciones.

Esto debemos realizar por cada elemento identificado.

Para identificar las tablas externas, podemos obtener con el siguiente query:

```
select directory_path||'/'||location Archivos
from dba_directories d, dba_external_locations l
where d.directory_name=l.directory_name;
```

Los objetos BLOB podemos obtenerlos con el siguiente query:

```
set serveroutput on;
set feedback off;
declare
type tcursor_ref is REF CURSOR;
v_cursor tcursor_ref;
v_sql varchar2(100);
v_bfile_loc bfile;
v_bfile_dir_name varchar2(30);
v_bfile_filename varchar2(250);
v_bfile_realpath varchar2(4000);
begin
for c in
(select owner,table_name,column_name
from dba_tab_cols where data_type='BFILE')
loop
dbms_output.put_line('External files en la columna: '||c.column_name||
```

```
', sobre la tabla: '||c.owner||'.'||c.table_name);  
v_sql:='select '||c.column_name ||' from '||c.owner||'.'||c.table_name;  
open v_cursor for v_sql;  
loop  
  fetch v_cursor into v_bfile_loc;  
  exit when v_cursor%notfound;  
  dbms_lob.filegetname(v_bfile_loc,v_bfile_dir_name,v_bfile_filename);  
  select directory_path into v_bfile_realpath  
  from dba_directories  
  where directory_name = v_bfile_dir_name;  
  dbms_output.put_line(v_bfile_realpath||'/'|| v_bfile_filename);  
end loop;  
close v_cursor;  
end loop;  
end;  
/
```

II. **Fase 2:** Migración de versión de Base de Datos

1. Ejecutamos los scripts de actualización hacia 11.2.0.3.

Configurando nuestro ORACLE_HOME al software de Oracle Database 10gR2 ejecutamos el siguiente script:

```
SQL> @utlu112i.sql
```

Esto va a permitir que la base de datos tenga registrada la misma versión de timezone que se encuentra en el software de Oracle Database 11gR2.

Este paso debe ser ejecutado con la base de datos en modo OPEN WRITE, la ejecución del script es de forma obligatoria.

Este script lo obtenemos en el ORACLE_HOME/dbs/admin del software de Oracle Database 11gR2.

También es importante que la memoria SGA se encuentre configurada al menos con 596 MB (412 MB es cuando migramos hacia Oracle Database 11gR2 x32 bits) previo a la actualización de la base de datos y asimismo corregir cualquier observación que devuelva la ejecución del script.

2. Procedemos a reiniciar la base de datos y la iniciamos en modo upgrade.

```
SQL*Plus: Release 11.2.0.3.0 Production on Thu May 17 17:56:58 2012
```

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

```
Connected to an idle instance.
```

```
SQL> startup upgrade  
ORACLE instance started.
```

```
Total System Global Area 768294912 bytes  
Fixed Size 2225448 bytes  
Variable Size 213912280 bytes  
Database Buffers 545259520 bytes  
Redo Buffers 6897664 bytes  
Database mounted.  
Database opened.
```

3. Realizamos el upgrade.

Ejecutamos el siguiente script: \$ORACLE_HOME/rdbms/admin/catupgrd.sql

Nota: Si el script entrega el siguiente error:

```
SELECT TO_NUMBER('MUST_HAVE_RUN_PRE-
```



```
UPGRADE_TOOL_FOR_TIMEZONE')
```

```
*
```

```
ERROR at line 1:
```

```
ORA-01722: invalid number
```

Como workaround debemos ejecutar el siguiente script:

```
SQL> update registry$database set TZ_VERSION = (select version FROM  
v$timezone_file);
```

```
SQL> commit;
```

4. Debemos reiniciar la base de datos en modo READ WRITE y ejecutamos el siguiente script: `$ORACLE_HOME/rdbms/admin/utlrp.sql`

Conclusión

En este artículo se demuestra la facilidad con que se puede migrar la base de datos Oracle entre diferentes plataformas con un tiempo menor de indisponibilidad y asimismo se entrega los pasos para actualizar de versión 10.2.0.4 a 11.2.0.3 con las mínimas consideraciones.

Recomiendo la revisión de las siguientes notas de My Oracle Support (MOS):

- How to transport a Tablespace to Databases in a Physical Standby Configuration [ID 467752.1].
- Transporting Tablespaces with Different Block Sizes Between Databases [ID 144212.1].
- Changing between 32-bit and 64-bit Word Sizes [ID 62290.1].
- Different Time Zone Version In Registry\$Database And V\$Timezone_file [ID 1255474.1].
- How to Transport a Tablespace from a 10g ASM on Windows to 11g ASM on AIX Using the DBMS_FILE_TRANSFER Package [ID 973484.1].
- Complete Checklist for Manual Upgrades to 11gR2 [ID 837570.1].

Publicado por Ing. Francisco Riccio. Es un IT Specialist en IBM Perú e instructor de cursos oficiales de certificación Oracle. Está reconocido por Oracle como un Oracle ACE y certificado en productos de Oracle Application & Base de Datos.

e-mail: francisco@friccio.com

web: www.friccio.com