Bank Expert Data Conversion Guide

Setting up Data into Oracle

Initiating the Conversion

Analyse the conversion of a module.

Important SQL Query techniques

Important PL-SQL techniques

Installing Oracle 10G

Installing SQL Developer

**Oracle SQL Plus:**

Login :system

Password: director

Note:

To Open Oracle pages in browser.

<http://localhost:5560/isqlplus/login.uix>

1. Creating a user in Oracle

CREATE USER <BANK\_CODE\_WITH\_BRANCH\_NO> identified by <PASSWORD>;

**E.g. CREATE USER SAIBABA\_SIGNIDENTIFIED BY DIRECTOR;**

2. GRANT DBA rights to this user.

GRANT DBA TO <user\_name>;

**e.g. GRANT DBA TO SAIBABA\_SIGN;**

Rather than giving specific access to the user, here we opt to give DBA rights to the schema. This will simplify the access control required by the schema.

3.CREATE TABLESPACE for above user

CREATE TABLESPACE <tables\_space\_name> DATAFILE ‘<path\_name>’ SIZE <size\_in\_mb> AUTOEXTEND ON NEXT <auto\_increment\_size>;

**e.g. CREATE TABLESPACE JANKALYAN\_OLD DATAFILE 'D:\TABLESPACE\JANKALYAN\_OLD.DBF' SIZE 50M AUTOEXTEND ON NEXT 10M;**

**Deleting Tablespace if required :**

first make it offline

**alter tablespace SAIBABA\_BANKoffline;**

then delete

**drop tablespace SAIBABA\_BANKincluding contents;**

Note :

-Keep the tablespace name same as SCHEMA.

- To find previous datafile use following query

**SELECT NAME,  
 FILE#,  
 STATUS,  
 CHECKPOINT\_CHANGE# "CHECKPOINT"   
 FROM V$DATAFILE;**

-Make sure that the drive where tablespace will be created has enough disk space available to accommodate years of input data.

4.CHANGE DEFAULT TABLESPACE

ALTER USER <schema\_name> DEFAULT TABLESPACE <tablespace\_name>;

**e.g. ALTER USER SAIBABA\_SIGN DEFAULT TABLESPACE SAIBABA\_SIGN;**

By default a newly created schema / user’s data will be put in the tablespace SYSTEM. So we need to change the default tablespace of the new user.

DROP ALL TRIGGERS IN USER:

select 'drop trigger ' || owner || '.' || trigger\_name || ';' from all\_triggers

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ORACLE SQL DEVELOPER :**

1.Open ORACLE SQL DEVELOPER

2.In Connections Tab: Right Click on connections and create new connection.

e.g.

connection name: VAIS01

user name: VAIS01

Password: director

hostname: localhost

port:1521

SID: orcl

3.Test the connection

4.Connect

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Data Importing:**

1. before importing data via dbf\_to\_oracle tool please make changes in E:\CSVTool\DATA\_IMPORT\DBF\_TO\_ORACLE\_V2.0\main\resources\db\_settings.txt

Change user name as VAIS01

2. Run DBF\_TO\_ORACLE\_V2.0.exe

check log file for errors: CSVTool\DATA\_IMPORT\DBF\_T0\_ORACLE\LOG\

**Run CSVTool:**

1. Make changes in CSVTool/INIT/INSERT\_CSV\_CONFIG
2. use CSVTool/EXEC.sql to run
3. For Bank level : @E:\CSVTool\INIT.SQL E:\CSVTool\
4. For Branch level : @E:\CSVTool\INIT\_JPNM01.SQL E:\CSVTool\
5. For Object Wise : @E:\CSVTool\MAIN.SQL E:\CSVTool\
6. Create CSV : @E:\CSVTool\JUSTCSV.SQL E:\CSVTool\

FOXBASE useful Commands:

**To run Photo and Signature script:**

1. Use utility “ImageSignRenamefinal”

To check output directories for sign and photos::

SELECT owner, directory\_name, directory\_path

FROM all\_directories;

Check for:

SIGN\_INPUT\_DIR

SIGN\_OUTPUT\_DIR

PHOTO\_INPUT\_DIR

PHOTO\_OUTPUT\_DIR

and create apropriate folders at path

1. Login as system

CREATE OR REPLACE DIRECTORY SIGN\_INPUT\_DIR

AS ‘D:\CSVTool\_SANT\DATA\_IMPORT\SIGNS';

CREATE OR REPLACE DIRECTORY SIGN\_OUTPUT\_DIR

AS ‘D:\CSVTOOL\_AJIT\_MULTI\OUTPUT\SIGN';

CREATE OR REPLACE DIRECTORY PHOTO\_INPUT\_DIR

AS ‘D:\CSVTool\_SANT\DATA\_IMPORT\PHOTOS’;

CREATE OR REPLACE DIRECTORY PHOTO\_OUTPUT\_DIR

AS ‘D:\CSVTOOL\_AJIT\_MULTI\OUTPUT\PHOTO';

CREATE OR REPLACE DIRECTORY ADDRESS\_PROOF\_OUTPUT\_DIR

AS ‘D:\CSVTOOL\_AJIT\_MULTI\OUTPUT\ADDRESS\_PROOF';

CREATE OR REPLACE DIRECTORY ID\_PROOF\_OUTPUT\_DIR

AS ‘D:\CSVTOOL\_AJIT\_MULTI\OUTPUT\ID\_PROOF';

GRANT READ, WRITE ON DIRECTORY SIGN\_INPUT\_DIR TO SNNSPH01;

GRANT READ, WRITE ON DIRECTORY SIGN\_OUTPUT\_DIR TO AJIT\_MULTI;

GRANT READ, WRITE ON DIRECTORY PHOTO\_INPUT\_DIR TO SNNSPH01;

GRANT READ, WRITE ON DIRECTORY PHOTO\_OUTPUT\_DIR TO AJIT\_MULTI;

GRANT READ, WRITE ON DIRECTORY ADDRESS\_PROOF\_OUTPUT\_DIR TO AJIT\_MULTI;

GRANT READ, WRITE ON DIRECTORY ID\_PROOF\_OUTPUT\_DIR TO AJIT\_MULTI;

1. Login as sys.

connect sys/director as sysdba;

**e.g. connect system as sysdba**

1. Set access to the schema.

GRANT EXECUTE ON fso\_pkg TO jaym01;

Compile Packages “SYS\_FSO\_PKG\_DEFN” and “SYS\_FSO\_PKG\_BODY”

1. Run scripts for sign and photo conversion.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. IMPORT THE DATA

**Importing the oracle data file.**

**Basic Syntax :**

**imp file**=<source\_file\_path>**log=**<log\_file\_path>**fromuser=**<source\_username>**touser=**<new\_user> commit=y ignore=y

Example :

1) Basic example

**impfile**=E:\MAIN.DMP LOG=E:\MAIN.LOG **fromuser**=ADMIN**touser**=MADHA**commit**=y **ignore**=y

Execute above command at dos prompt. This will prompt you to enter the username, password. Provide the username specified to **touser** parameter, and corresponding password.

Note :

-fromuser parameter value must match to the source data username. In above example ORA\_DATA.DBF file is owned by the user SB\_SHIV.

-Soure file may also have the extension **“.dmp”**.

**Check database size :**

select (select sum(bytes)/1024/1024 from dba\_data\_files)+

(select sum(bytes)/1024/1024 from dba\_temp\_files) "Size in MB" from dual;

**EXPORTING ORACLE DATA ---**

1. Open CMD as administrator
2. write following command in cmd.exe

**Syntax :**

**C:\**exp %schema%/%pwd%@%sid% FILE=%user%\_%dd%\_%mm%\_%yy%.dmp log=%user%\_%dd%\_%mm%\_%yy%.log  
compress=N rows=Y grants=Y buffer=500000 FULL=Y statistics=NONE

**Example : C:\**exp **sahyadri**/director@orcl FILE=**D:\Exported\sahyadri.dmp** log=**D:\Exported\sahyadri**.log statistics=NONE

**if error message comes then use following command:**

**C:\ exp SAIBABA\_BANK/DIRECTOR@orcl FILE=D:\Exported\SAIBABA\_BANK.dmp log=D:\Exported\SAIBABA\_BANK.log**

11 G TO 10G EXPORT CONVERSION:

**expdpscott/tiger version=10.2.0.3.0 directory=DUMPDIR dumpfile=DUMPFILE.dmp logfile=DUMP.log**

**Import Oracle data(.DBF) :**

**Imp file=**E:\FINAL\_JANTA\_BANK\WEDNESDAY.dmp LOG=E:\FINAL\_JANTA\_BANK\WEDNESDAY.LOG**fromuser=**ebankdelight**touser=JANATA\_BANK20** commit=y ignore=y

**impfile**= E:\FINAL\_JANTA\_BANK\ WEDNESDAY.DMP LOG=E:\FINAL\_JANTA\_BANK\ WEDNESDAY.LOG **fromuser=**ebankdelight**touser=JANATA\_BANK20** GRANTS=NO

11 G TO 10G IMPORT CONVERSION:

impdp JANATA\_BANK20/DIRECTOR@orcl DIRECTORY=FINAL\_JANTA\_BANK REMAP\_SCHEMA=ebankdelight:JANATA\_BANK20 dumpfile=WEDNESDAY.DMP logfile=JANATA\_BANK20.log

12c TO 10G IMPORT CONVERSION:

impdp JANKALYAN/DIRECTOR@orcl DIRECTORY=JANKALYAN\_SOURCE\_DB REMAP\_SCHEMA=C##JANKALYAN:JANKALYAN dumpfile=JANKALYAN.DMP logfile=JANKALYAN.log

Import specific tables

**impfile=** E:\JANTA\_BANK\_DB\MONDAY.DBF **log=**E:\JANTA\_BANK\_DB\ MONDAY.log **fromuser=**ebankdelight**touser=**JANATA\_BANK**TABLES**=(PIGMY\_CUMULATIVE\_TRANS) commit=y ignore=y

**Various values for TABLES parameter :**

i.Comma separated list of table names :

e.g. **TABLES=** (ACMST,ALMST,ASMST,CLMST,FDMST,FTMST,INMST,SIMST,IMAGE)

ii Use of Wild card characters.

e.g. **TABLES=(SIMST,IMAGE, T\_%)**

v**Importing sql server database.**

ØCreate a connection for the MS SQL Server database in SQL Developer

ØAfter successful connection right click the table objects in this SQL Server connection and click on Copy.

ØIf copy fails resolve the problem and then retry.

v**Importing other non-conventional** (other than oracle / sql server) **database.**

This may require writing customized tool to bring the data into oracle database.

e.g. In case of Vaishya Sahakari Bank, we developed a tool in JaVA which imported **Foxbase - .dbf** files to oracle.

6. ORACLE DIRECTORY SETTING

CREATE OR REPLACE DIRECTORY <directory\_name> AS ‘<directory\_path>’;

e.g.

- Login to the desired schema.

- CREATE OR REPLACE DIRECTORY CSV\_OUTPUT\_DIR AS ‘D:\CSVTOOL\_LOKMANGAL\OUTPUT';

- GRANT READ, WRITE ON DIRECTORY CSV\_OUTPUT\_DIR TO LOKMANGAL;

Note :

-Issue read, write grant to the schema under consideration.

-Similarly we can create PHOTO\_OUTPUT\_DIR, SIGN\_OUTPUT\_DIR to store photos, signature images respectively.

**Verify the directory path using the following command**

**SELECT owner, directory\_name, directory\_path**

**FROM all\_directories**

-Drop a directory using the command :

DROP DIRECTORY <directory\_name>;

-Dropping a directory and issuing “CREATE DIRECTORY ….” Statement OR reissuing the “CREATE OR REPLACE DIRECTORY …” statement with changed path, will recreate or change the directory reference.

7. UTL\_FILE ACCESS

To be able to write **.txt** log files or **.CSV** output files to the disk, the user/schema must have access permission to UTL\_FILE. This can be done with following statement.

-Login as sys.

**e.g. connect sys/director as sysdba;**

-Set access to the schema.

GRANT EXECUTE ON UTL\_FILE TO <user\_name>;

**e.g.**

**GRANT EXECUTE ON UTL\_FILE TO SNNSPH01;**

**GRANT EXECUTE ON UTL\_FILE TO ajit\_multistate;**

**Initiating the Conversion**

Follow EXEC.sql :

1. INIT.sql
2. INIT\_branch01/02/03/04/05/06..sql
3. MAIN.sql

1.Run **INIT**.sql :

This process includes executing the script **init.sql,** which in turn calls execution of various sql scripts that create structure, build data, create functions required for processing the data of a branch.

A general file INIT.SQL contains following sequence of commands:

|  |  |  |
| --- | --- | --- |
| **#** | **Script file** | **Description** |
| 1 | CREATE\_GLMAP.sql | Create structure for GLMAP table |
| 2 | CREATE\_NEW\_BEACNO.sql | There might be a need to map overflowing account no. to valid account no. To hold such records NEW\_BEACNO table is created. e.g. Saving account with Acc. No. 99999 has to be renamed to current max account no. 8727 or any available account no. less than the current max account no. |
| 3 | INSERT\_NEW\_BEACNO | Script for inserting record into NEW\_BEACNO |
| 4 | CREATE\_CSV\_CONFIG | Create a configuration table which contains parameters for each branch. |
| 5 | INSERT\_CSV\_CONFIG | Script for inserting record into CSV\_CONFIG |
| 6 | CREATE\_CSV\_MASTER | Build structure to hold different conversion objects. |
| 7 | INSERT\_CSV\_MASTER | Add records to CSV\_MASTER table |
| 8 | CREATE\_BE\_CODES\_MAP | Construct structure to hold the mapping of various codes. e.g. Constitution Code mapping between client codes and Bank Expert codes |
| 9 | INSERT\_BE\_CODES\_MAP | Insert mapping records into BE\_CODES\_MAP table. |
| 10 | FUN\_GET\_ACTIVE\_BRANCH\_CODE | Find out the branch from CSV\_CONFIG, whose data has to be converted currently |
| 11 | FUN\_GET\_BE\_ACTNO | Compute the account no.  ( Branch code + GL No. + Account No) |
| 12 | FUN\_GET\_BE\_CODE | Get Bank Expert code for client codes. |
| 13 | FUN\_GET\_BE\_CONST\_CODE | Get constitution code |
|  |  |  |

Apart from these regular initiation scripts, there might be other scripts required which, can be added to **init.sql**.

2.Branch Initialization

**Get all tables list and number of records:**

SELECT SUBSTR(TABLE\_NAME,1,2),TABLE\_NAME, NUM\_ROWS FROM USER\_TABLES WHERE NUM\_ROWS > 0

ORDER BY TABLE\_NAME;

**Get all tables and its column names:**

SELECT

SUBSTR(TABLE\_NAME,1,2),TABLE\_NAME, COLUMN\_NAME, DATA\_TYPE,

DATA\_LENGTH, DATA\_PRECISION,

DATA\_SCALE, DATA\_DEFAULT

FROM USER\_TAB\_COLS WHERE TABLE\_NAME IN

( SELECT TABLE\_NAME FROM USER\_TABLES WHERE NUM\_ROWS > 0 )

ORDER BY TABLE\_NAME, COLUMN\_NAME;