

Step by step guide.

Database migration using Wizard, Studio and
Commander.

Based on migration from Oracle to PostgreSQL(Greenplum)

Version 1.0

Contents

SQLWays Wizard

Welcome page.....	3
Choose a Source Database.....	3
Choose a Target Database.....	4
Specify Database Objects or Query.....	5
Specify DDL(Data Definition Language) and Data Options.....	6
Specify Export File Options.....	9
Specify Import Options.....	10
Review the selections.....	11
Migration Status.....	11

SQLWays Studio

SQLWays Commander

Export Directory

Log files.....	16
Batch files (for SQLWays Wizard only).....	17
DDL and Data Files.....	18

SQLWays Wizard

SQLWays Wizard was created to convert different databases using ODBC connection. This tool contains various options that help you in tuning the migration process to fit your needs as much as possible.

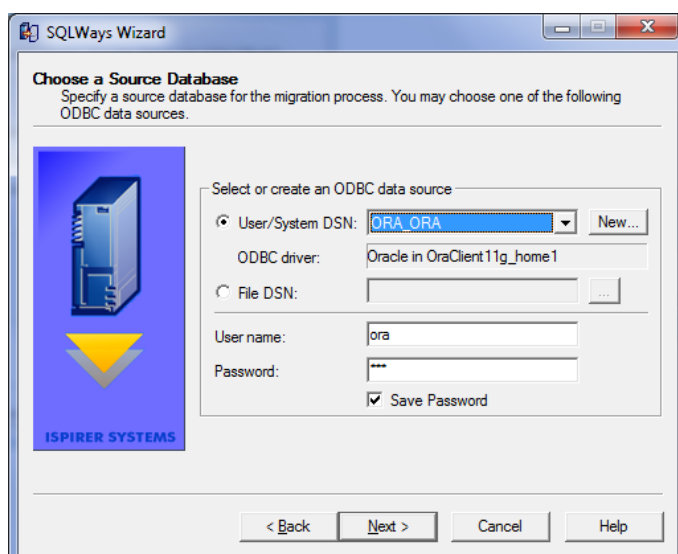
Welcome Page



This is a Welcome Screen that you can see when just start the SQLWays Wizard. At this page there is an option “Select a Project Directory or Create a New Project”. All information about your current conversion, like data type mapping, objects name mapping and etc., is saved in the folder that specified in this option. When the first conversion is complete you can specify existing project, so all options will be restored automatically.

Choose a Source Database

When you press “Next >” you can see page similar to below:



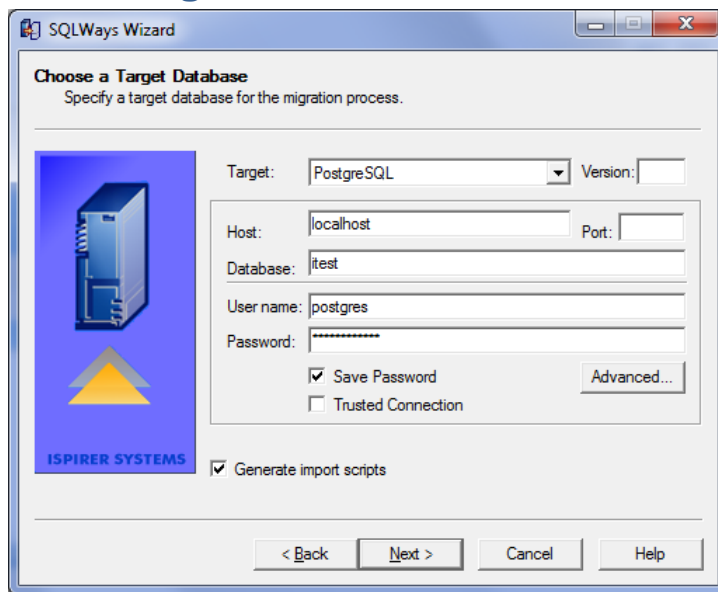
On this page we need to specify User or System DSN that correctly tuned up to the Source Database. By pressing on the button “New...” you can create a new one DSN to connect to the Database. Also you can choose file DSN, if you store you ODBC connection settings in the file.

Also you need to specify user name and password to connect to your database.

Checkbox “Save Password” is needed to store password for next session, if you plan to use current project more than once.

When you press “Next>” SQLWays tries to connect to the source database and read information about objects that are located in it.

Choose a Target Database



First of all we need to specify Target Database from the dropdown list. Also you can specify version of the target database. Some conversion may differ depending on the version specified.

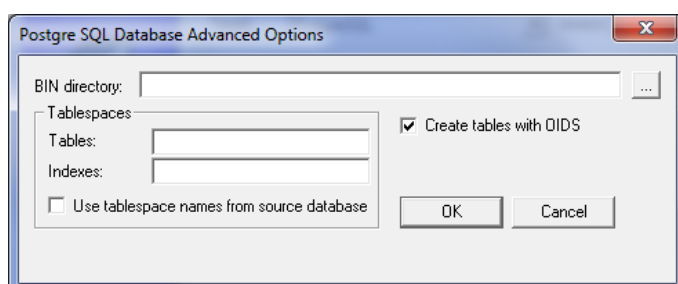
The rest information is required to connect to the database when import of the converted objects begins.

HOST - is name of the server where target database is located.

PORT – port that used to connect to the database.

User name and password – credentials that are required to logon to the target database.

When you press “Advanced...” button you can see the following screen:



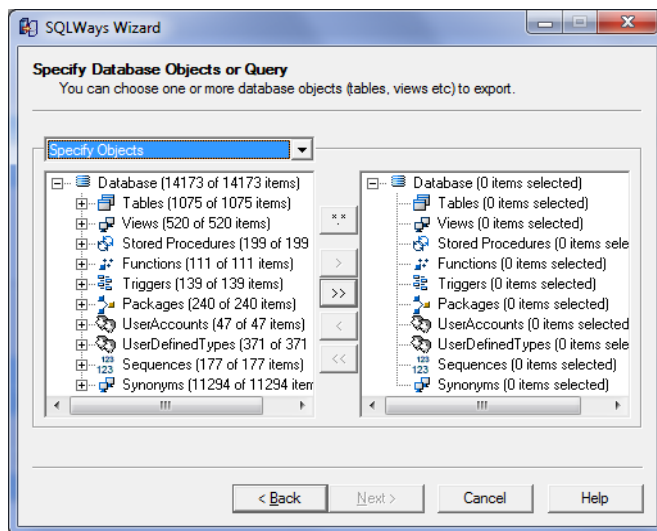
In “Bin directory” you need to specify the path to the Native client tool of the Target Database. So our tool works with the Source Database using the ODBC connection and with Target Databases using their Native tools. In most cases our tool can identify path to the Native tool itself from the Registry of your System. But we recommend to specify this path in any case.

Rest of the options in this page are specific for each database and intuitively simple.

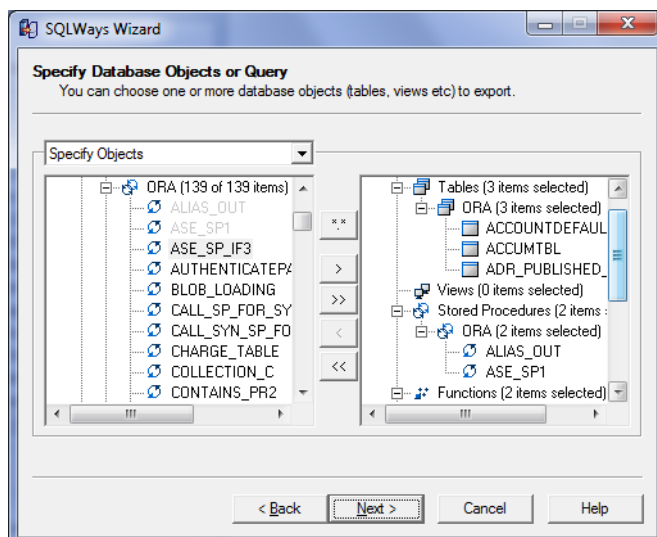
This is the main option on this page.

Specify Database Objects or Query

On this page at the left pane you can see all objects that exist in the source database, such as tables, views, stored procedures and etc.

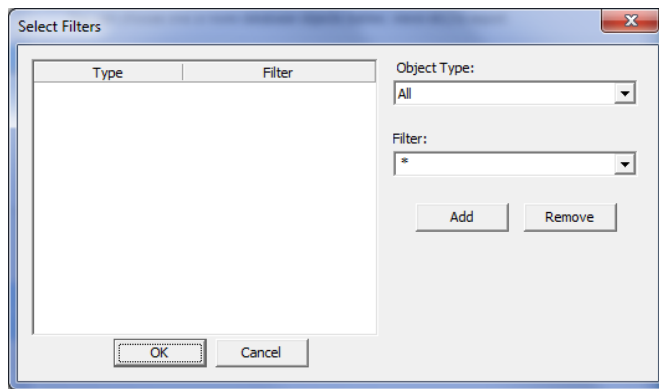


At the right pane you will see objects that need to be converted .



If, at the beginning you have specified existing SQLWays project, objects that were selected during previous session will be depicted automatically in the right pane.

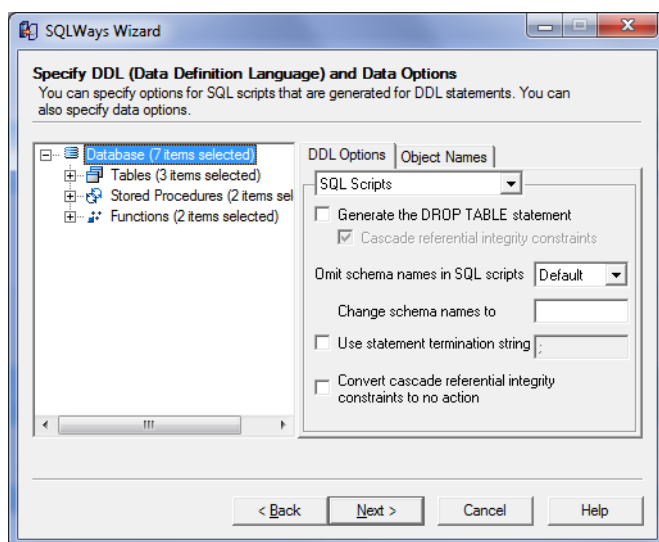
Also in this page you can specify filter in order to depict the required for this particular migration objects only. It is located under the button “*.*”.



In this box you can add several filters for different database objects. To add filter press button “Add”, to remove – button “Remove”. If the filter is added in the left pane you will see objects that suit to the added filter only and we can select objects for conversion from them.

Specify DDL(Data Definition Language) and Data Options

On this page you can specify options for the selected objects to tune up your conversion to your requirements. In this Guide only a few of them are described, those that are most commonly used.

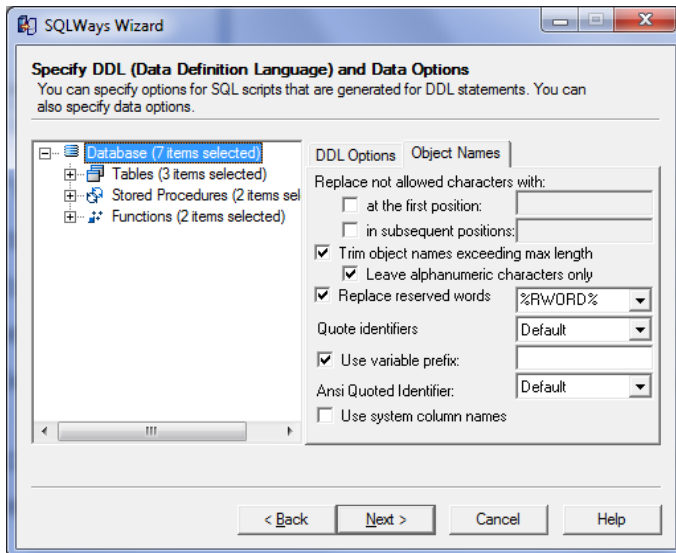


First option, that we would like to describe is “Omit schema names in SQL scripts”. This option has 3 values: “Yes”, “No”, “Default”. When option is set to “Yes”, schema names will be trimmed off from every occurrence in the result scripts (from stored procedures, functions, table names and etc.) . If option set to “No”, schema name will be left before the object name or added default schema name (only for MSSQL Server). If option is set to “Default” behavior is depends on the target Database.

If option “Omit schema names in SQL scripts” is specified as “No” or “Default”, another option becomes available – “Change schema name to”. This option changes all schema names to a particular schema that is specified in the text box.

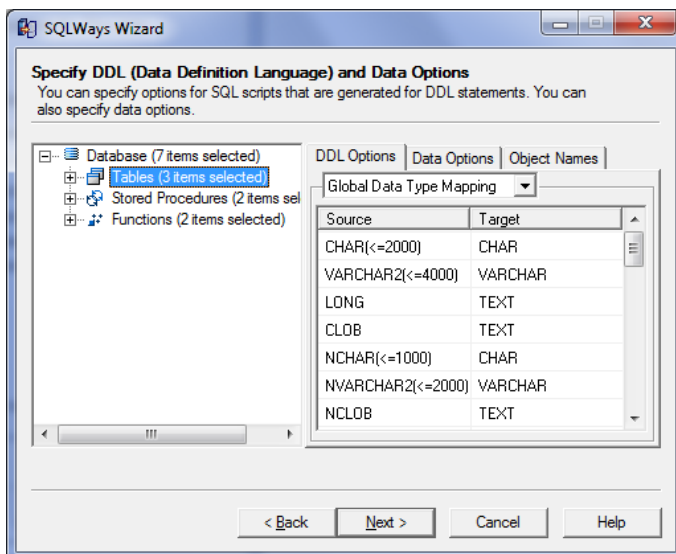
One more useful option is “Generate DROP TABLE statement”. This option is unchecked by default. When it is checked, for each table in the converted database there is added drop statement. This is useful when you’re not sure if this table exists or not, or if you want to re-import the DDL of the table.

Now let’s click the tab “Object Names”.

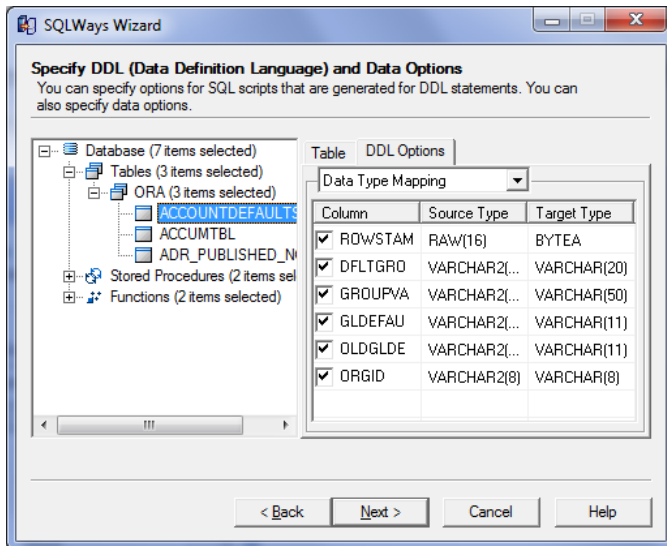


On this tab please note the option “Replace reserved words”. Our tool can handle situations when one word in the source database is not reserved and in target database is reserved. In this case such word is delimited with double quotes. Using option “Replace reserved words” you can change this behavior and specify for example “r_%RWORD%”, so all reserved word in this case will be preceded with “r_”.

Now let’s choose “Tables” node in the left pane. You can see the options like in the picture below.



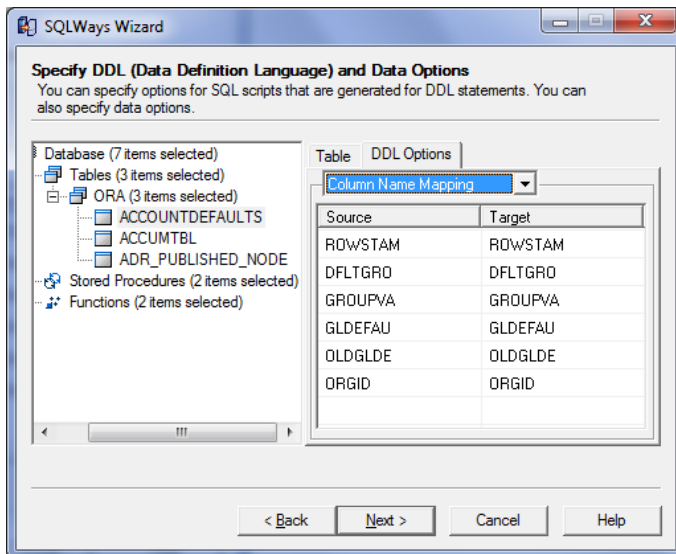
In SQLWays we have two types of mapping first is “Global Data Type mapping” and second is “Local Data Type mapping”. When you select “Tables” in the left pane we can change “Global Data Type mapping”. Conversion of each type can be changed for the whole conversion. To see the “Local Data Type mapping” you need to choose one of the tables like in the picture below and click the tab “DDL options”.



You can see the list of the columns with appropriate data types. At this tab we can change conversion of every column, or even exclude the column from conversion.

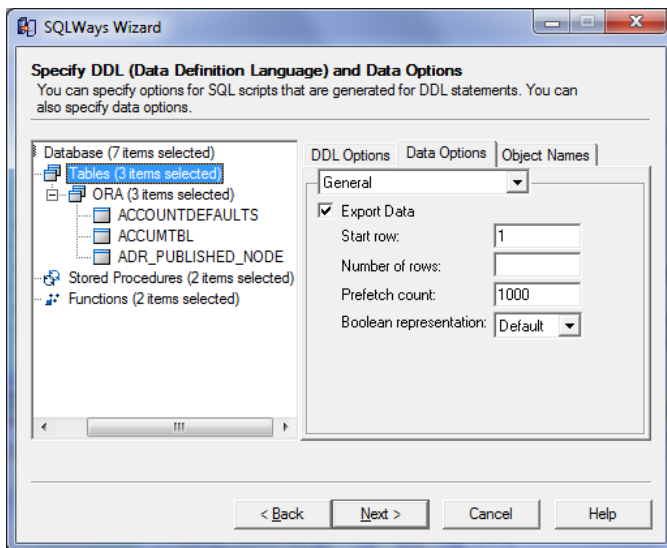
Please notice that changes made to “Local Data Type mapping” has a higher priority than changes made to “Global Data Type mapping”.

Also you can change column names if you change “Data Type Mapping” from the drop down list to “Column Name Mapping” like in the picture below.

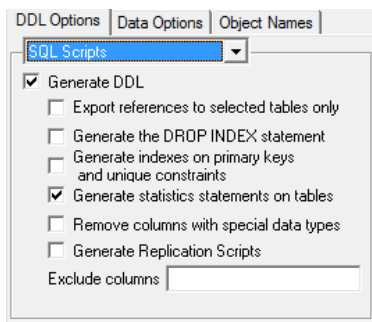


Back to “Tables” and select tab “Data Options”. Here you can see some additional options that correspond to data migration.

Main option here is “Export Data”. This option is useful when you don’t want to convert and transfer data. If the option unchecked data will not be converted, only DDL will be converted.

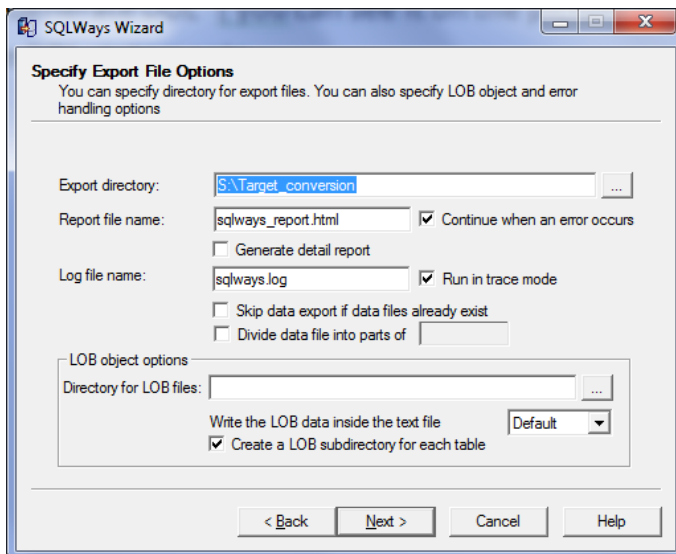


There is another one option “Generate DDL” (you can see it on the picture below) that allows migrating only data without any conversion of the DDL statements.



This is the main option in the page.

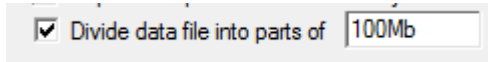
Specify Export File Options



First option is “Export Directory”. This option contains path to the folder where files, generated by SQLWays, are placed after conversion. This is useful for importing files to the target database. You can start import from that directory any time.

Next useful option is “Run in trace mode”. This option generates additional information about conversion in the file sqlways.trc (this file is useful for our developers) and adds commented source of the converted object into result file.

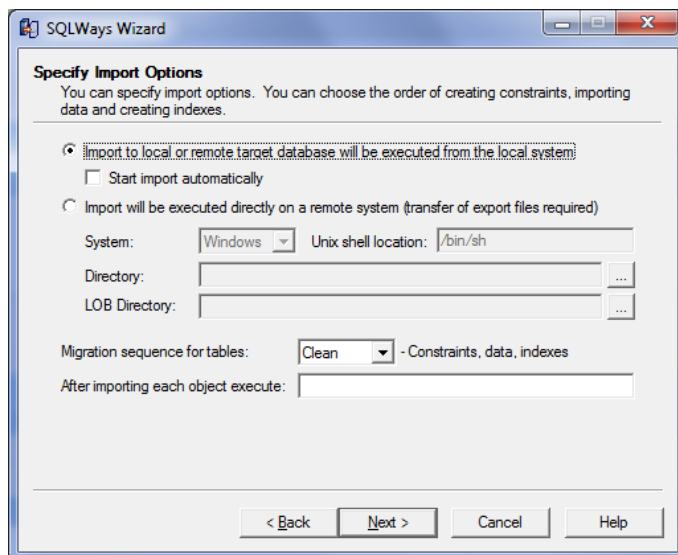
Next option “Divide data file into parts. This option is useful when the generated data file is too large to be maintained by OS, so it can be divided into parts.



For example, we can divide one large data file that originally contains several G of data, into appropriate number of files with size 100M.

Specify Import Options

This page corresponds to import options. Here you can specify where import is executed, OS of the remote computer (if import is executed directly on the remote system) or migration sequence.



Let's check every option on this page more deeply.

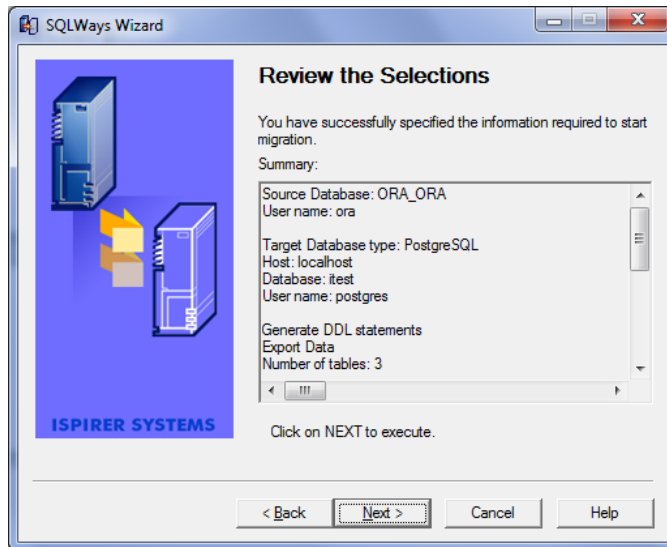
First option is “Import to local or remote target database will be executed from the local system”. This option is used if you want to migrate DDLs and data directly from your local machine to the target database. This option is set as default. In pair with this option you can check option “Start import automatically”. If this option is checked, import starts right after the export and conversion of all selected objects.

Next option “Import will be executed directly on a remote system (transfer of export files required)”. This option must be chosen if you want to start import from another machine. In this case you can choose OS of the remote machine and if it's a UNIX system specify path to shell. Also in this case you must specify path to export directory on the remote system, where converted files are transferred. Also you can specify LOB directory.

Notice – if you choose to start import from the remote machine, you have to transfer converted files on this machine manually.

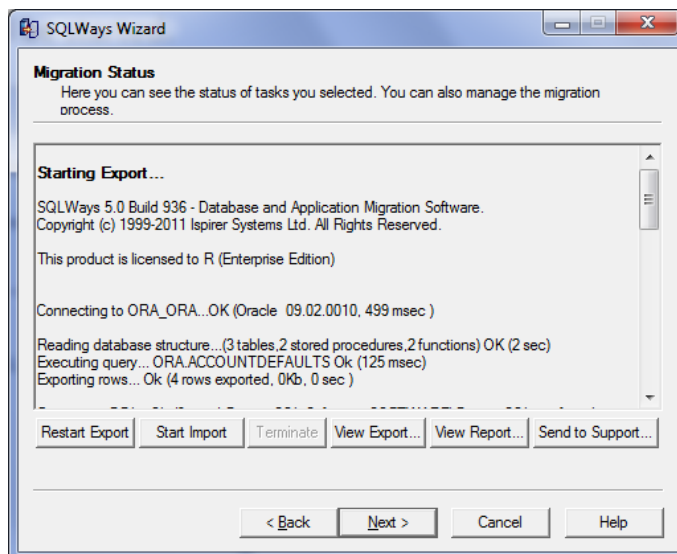
Review the selections

On this page you can see summary information regarding your conversion right before the export. After clicking “Next” button, the export starts.



Migration Status

On this page you can watch on the status of the conversion and results right after export.



If you have checked option “Start import automatically” you can also check status of the import. If not, just click on the button “Start Import” to start import manually.

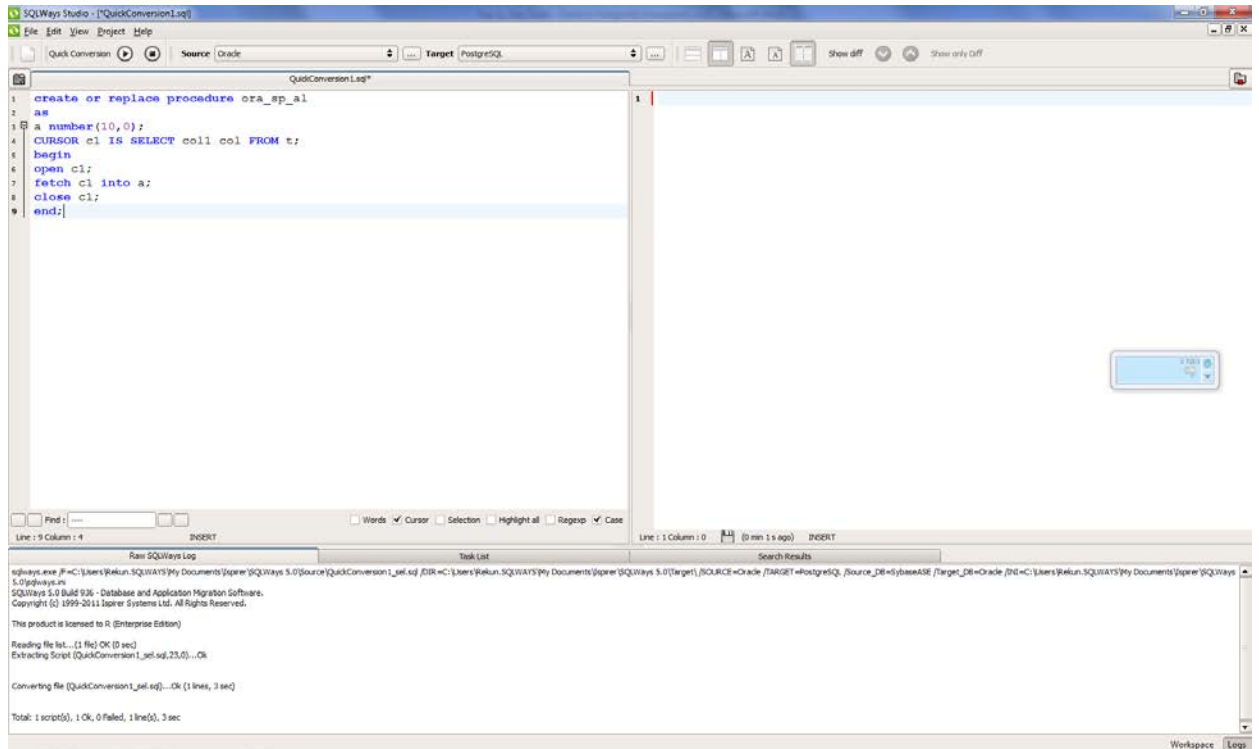
Button “View Export...” opens export folder with the converted files.

Button “View Report...” opens a short report in your browser to review results of the conversion.

Button “Send to Support...” generates a template of q letter with attached sqlways.log and sqlways_imp.log files that can be sent to our Support Team.

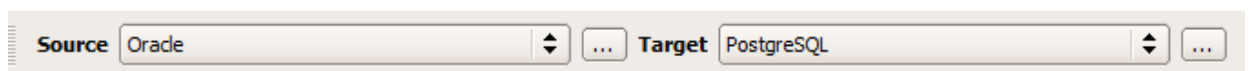
SQLWays Studio


This tool is actually used for script conversion. Commonly it is used for single script conversion for checking results of the conversion.



In Studio you can open existing file or create new for Quick Conversion and copy into blank file some statements or procedures or something else.

To convert in the SQLWays Studio, first of all, you need to specify source and target Databases from the corresponding dropdown lists:



Source script should be placed at the left pane and target will appear at the right pane. Then press button  to start the conversion.

At the bottom pane you can see the whole process of the conversion and the results (was it successful or not).

Some useful information can be seen in this pane.

First of all, the first row show you which command line was used to run the conversion. Then every script that was converted has its own status – was it converted normally or with errors.

And the last - Total information. For example when we convert script from above picture we will get next total information about conversion:

Total: 1 script(s), 1 Ok, 0 Failed, 8 line(s), 4 sec

That's mean that we have one script in this session that converted normally. It contains 8 lines and was converted in 4 seconds.

One more useful feature in the Studio is that you can convert only selected statement. For example, we will select only SELECT statement from the converted procedure and SQLWays studio will convert only it.

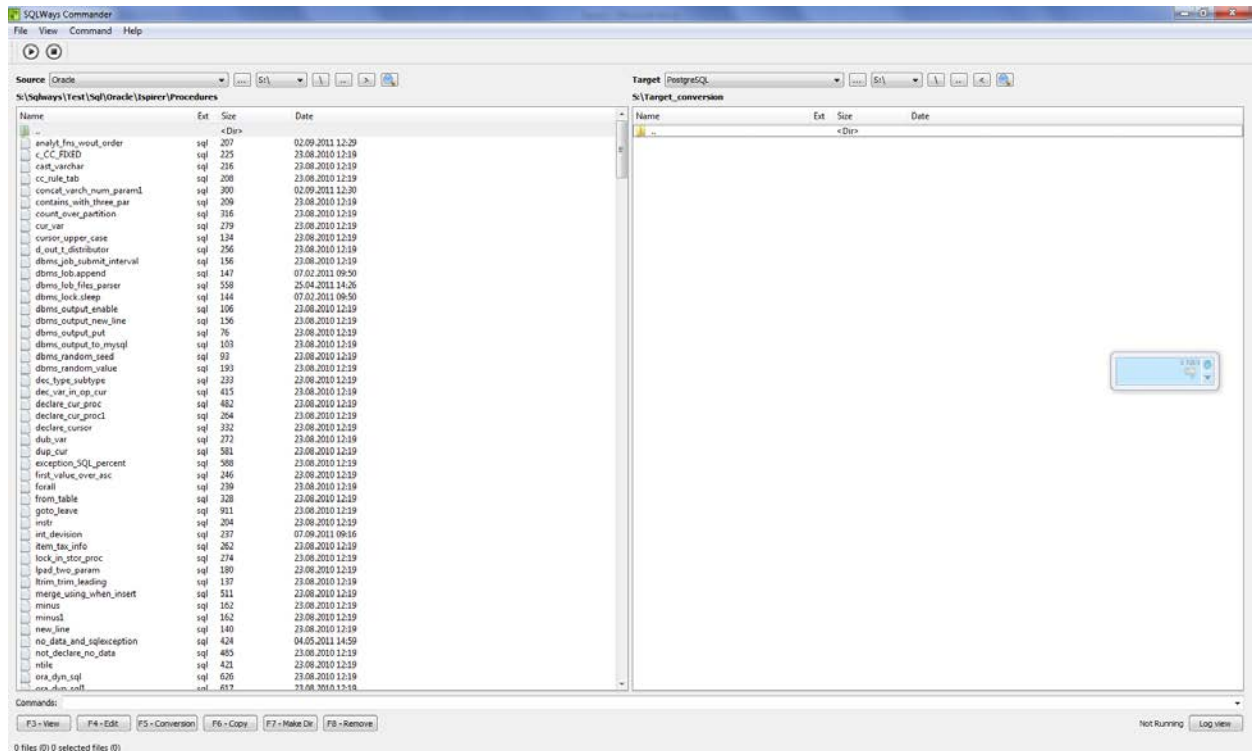
```
create or replace procedure ora_sp_a1
as
a number(10,0);
CURSOR c1 IS SELECT col1 col FROM t;
begin
open c1;
fetch c1 into a;
close c1;
end;
```


```
1 SELECT col1 AS col FROM t;
```

In summary Studio is mainly purposed to convert single scripts to provide quick check of the results to customer.


SQLWays Commander

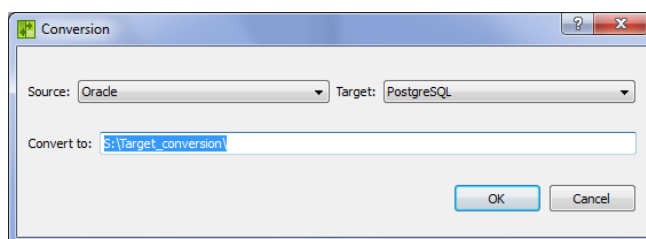
Main purpose of this tool is to convert a set of scripts, or folder, or both of them.



Conversion options are set like in the Studio. In the same way we need to specify source and target databases and start the conversion. Main feature of this tool that it can convert multiply files. We can select one, two or more files in any order and just press button  to start the conversion.

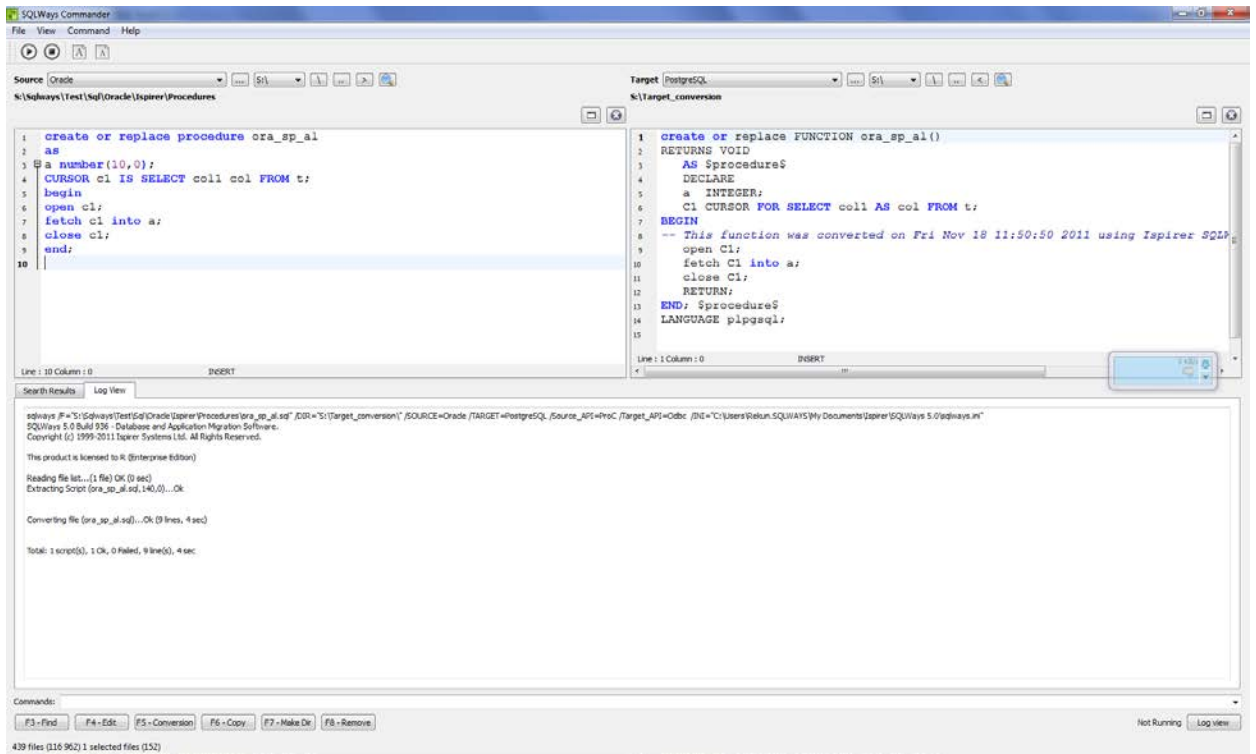
Notice that source files should be always at left pane. Right pain usually works as folder where results placed.

When we press button  to start the conversion next window appears:

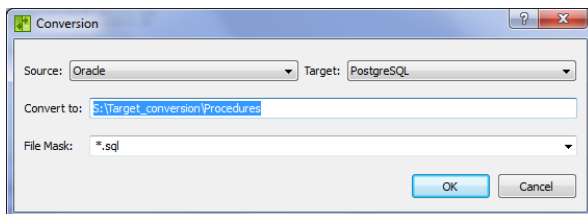


Here we confirm Source and Target Databases and destination folder.

When conversion is finished you can just click on the result to check it. It will be opened like a text file in the appropriate pane. The same way you can see source file.



Like we convert the script we can convert the whole folder. We just need to select the folder and start the conversion. The confirmation window in this case will have one addition option – file mask. In this option you can specify what mask should apply our tool to select appropriate files for conversion. For example, if we want to convert all files we need to specify *.* mask, but if we want to convert all files with extension sql , we need to specify *.sql mask.



So, main purpose of SQLWay's Commander is to convert multiply files or folder with the filter mask.

Export Directory

After migration, in export folder there will be generated a lot of files with different extensions and purposes. These files are very useful for checking results of export and import.

In this chapter we will describe their meaning.

After export step is complete in the export folder you will get files divided into following groups:

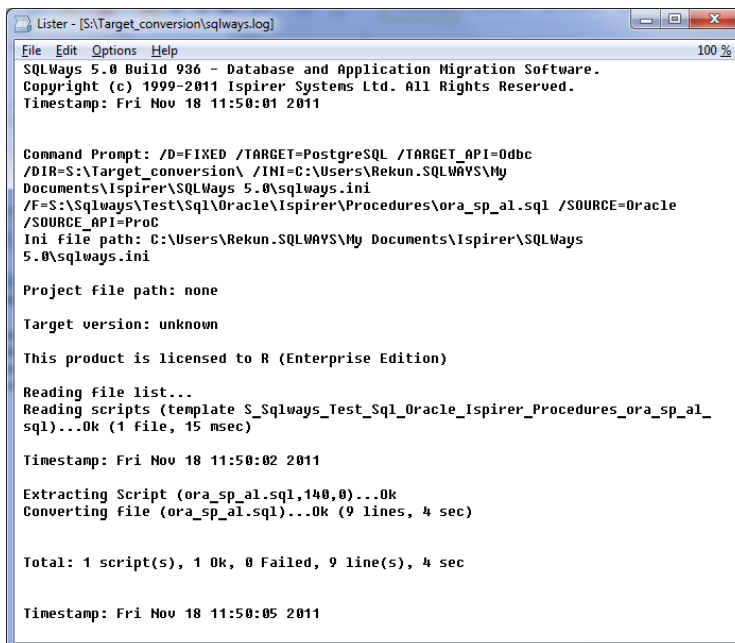
Log files, batch files, ddl and data files.

Log files

In this group you will have three files (if you complete required steps.).

First file – sqlways.log

This file contains basic information about export stage and it's created after this stage.



```
Lister - [S:\Target_conversion\sqlways.log]
File Edit Options Help 100 %
SQLWays 5.0 Build 936 - Database and Application Migration Software.
Copyright (c) 1999-2011 Inspirer Systems Ltd. All Rights Reserved.
Timestamp: Fri Nov 18 11:50:01 2011

Command Prompt: /D=FIXED /TARGET=PostgreSQL /TARGET_API=odbc
/DIR=S:\Target_conversion\ /INI=C:\Users\Rekun.SQLWAYS\My
Documents\Inspirer\SQLWays 5.0\sqlways.ini
/F=S:\Sqlways\Test\Sql\Oracle\Inspirer\Procedures\ora_sp_al.sql /SOURCE=Oracle
/SOURCE_API=ProC
Ini file path: C:\Users\Rekun.SQLWAYS\My Documents\Inspirer\SQLWays
5.0\sqlways.ini

Project file path: none

Target version: unknown

This product is licensed to R (Enterprise Edition)

Reading file list...
Reading scripts (template S_Sqlways_Test_Sql_Oracle_Inspirer_Procedures_ora_sp_al
.sql)...0k (1 file, 15 msec)

Timestamp: Fri Nov 18 11:50:02 2011

Extracting Script (ora_sp_al.sql,140,0)...0k
Converting file (ora_sp_al.sql)...0k (9 lines, 4 sec)

Total: 1 script(s), 1 0k, 0 Failed, 9 line(s), 4 sec

Timestamp: Fri Nov 18 11:50:05 2011
```

In this file you can receive next information:

Version of SQLWays that was used for conversion, command line used to start the conversion, information about any errors if they appear during conversion and total information about export stage.

We recommend checking log file after every export to identify if the error arose or no.

Next file – sqlways_imp.log.

This file contains information about import stage and it's created after this stage.

```
Listner - [\\ws028\vs$\support\Demo\GreenPlum\Oracle2PostgreSQL\Export\sqlways_imp.log]
File Edit Options Help 100 %
5.0 Build 936 - Database and Application Migration Software.
Copyright (c) 1999-2011 Inspire Systems Ltd. All Rights Reserved.
Timestamp: Wed Nov 02 19:10:31 2011

Importing database objects to PostgreSQL
Specifies Password
Create plpgsql language
psql:create_lang.sql:1: ERROR: language "plpgsql" already exists
Importing tables ordered by referential constraints (5 tables)

Importing table T to PostgreSQL
Creating table and its constraints using the PostgreSQL command line utility
Importing data using the PostgreSQL COPY command

Importing table TAB_PG_COPY to PostgreSQL
Creating table and its constraints using the PostgreSQL command line utility
Importing data using the PostgreSQL COPY command

Importing table TAB_SYN to PostgreSQL
Creating table and its constraints using the PostgreSQL command line utility
psql:tab_syn_ddl.sql:8: NOTICE: drop cascades to view view_tab_syn
Importing data using the PostgreSQL COPY command

Importing table TB_DATE to PostgreSQL
Creating table and its constraints using the PostgreSQL command line utility
psql:tb_date_ddl.sql:8: NOTICE: drop cascades to view vw_tb_date
Importing data using the PostgreSQL COPY command

Importing table T_WITH_IDX to PostgreSQL
Creating table and its constraints using the PostgreSQL command line utility
Importing data using the PostgreSQL COPY command
Creating indexes using the PostgreSQL command line utility
Importing views ordered by dependency (2 views)
Importing view VIEW_TAB_SYN to PostgreSQL
Importing view VW_TB_DATE to PostgreSQL
Importing sequences ordered by dependency (2 sequences)
Importing sequence ECMS_CERT_HEADERS_SEQ to PostgreSQL
Importing sequence ORA_SEQ_MAXVALUE to PostgreSQL
Importing stored procedures ordered by dependency (1 stored procedure)
Importing stored procedure SP_WITH_OUT_TYPE to PostgreSQL
psql:sp_with_out_type.sql:19: NOTICE: type reference t.col1%TYPE converted to
double precision
Importing packages ordered by dependency (1 package)
Importing package ORA.PKG_TEST to PostgreSQL
```

In this file you can find the following information:

Version of SQLWays that was used for migration; information about every imported object and, if the error is received, information about the error.

Third file – sqlways.trc

This file contains detailed information about export stage. This file generated only if option “Run in Trace mode” on the page “Specify Export File Options” is checked.

Information from this file is very useful for our developer team to identify complex issues in our tool.

Batch files (for SQLWays Wizard only)

If the option “Generate import scripts” on the page “Choose a Target Database” is checked, after conversion in export directory there will be generated batch files with appropriate commands to import every object. Also there will be created general batch file named “sqlways_all.bat” that starts all other files.

For example, below you can see one of the batch files that contain command to run sql file with create procedure statement.

```
Listner - [\\ws028\vs$\support\Demo\GreenPlum\Oracle2PostgreSQL\Export\create_lang.bat]
File Edit Options Help 100 %
p:\DBCLIENTS\PostgreSQL\8.2\bin\psql.exe -h localhost -U postgres -d itest
--file create_lang.sql >> crt_plpgsql.log
```

To start import for all objects manually you can run “sqlways_all.bat” file. It will start import of all specified objects.

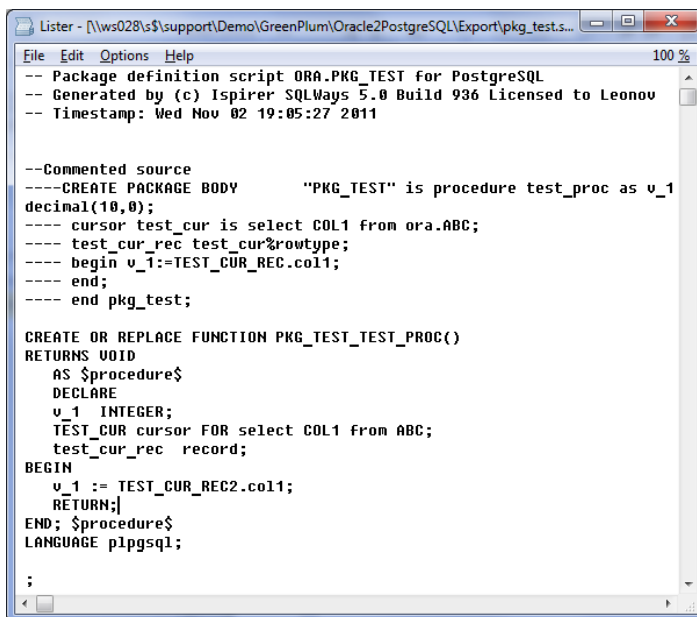
Also you can start both export and import from Export folder. To do this just run file "sqlways_exp_imp.bat"

DDL and Data files

DDL and Data files are the main files for import. They can be with three different extension *.sql, *.ldi(for SQLWays Wizard only), *.txt (for SQLWays Wizard only).

SQL files contain definition of the SQL statement that are executed during import. If the option "Run in trace mode" is checked in addition to SQL statements there is generated source SQL statement as comment. In this case you can compare source and target statement to check if the conversion works normally.

For example, below picture shows create procedure statement with commented source:



```
Listner - [\\ws028\vs$\support\Demo\GreenPlum\Oracle2PostgreSQL\Export\pkg_test.s...
File Edit Options Help 100 %
-- Package definition script ORA.PKG_TEST for PostgreSQL
-- Generated by (c) Ispirer SQLWays 5.0 Build 936 Licensed to Leonov
-- Timestamp: Wed Nov 02 19:05:27 2011

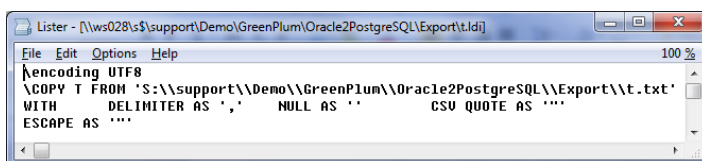
--Commented source
----CREATE PACKAGE BODY      "PKG_TEST" is procedure test_proc as v_1
decimal(10,0);
---- cursor test_cur is select COL1 from ora.ABC;
---- test_cur_rec test_cur%rowtype;
---- begin v_1:=TEST_CUR_REC.co11;
---- end;
---- end pkg_test;

CREATE OR REPLACE FUNCTION PKG_TEST_TEST_PROC()
RETURNS VOID
AS $procedure$
DECLARE
v_1 INTEGER;
TEST_CUR cursor FOR select COL1 from ABC;
test_cur_rec record;
BEGIN
v_1 := TEST_CUR_REC2.co11;
RETURN;|
END; $procedure$
LANGUAGE plpgsql;

;
```

LDI file contain command to start data load into target DB.

For example, on the picture below you can see command that start load data into PostgreSQL DB.



```
Listner - [\\ws028\vs$\support\Demo\GreenPlum\Oracle2PostgreSQL\Export\l.tldi]
File Edit Options Help 100 %
\encoding UTF8
\COPY T FROM 'S:\\support\\Demo\\GreenPlum\\Oracle2PostgreSQL\\Export\\t.txt'
WITH DELIMITER AS ',' NULL AS '' CSV QUOTE AS ''''
ESCAPE AS ''''
```

TXT files contain data that are loaded into target DB using command from *.ldi file.