

IBM Database Conversion Workbench 4.5.1

Teradata Quick Start Guide

Version: 1.1

Last Updated: Nov 6, 2020



Table of Contents

1.	INTRODUCTION.....	3
2.	TABLE & VIEW SUPPORT IN DCW	4
1.	CREATE TERADATA PROJECT	4
2.	IMPORT SQL FILE	5
3.	EVALUATE COMPATIBILITY	6
4.	CONVERT THE CODE	9
5.	RUN SQL	11
6.	SUPPRESS COMMENTS.....	11
7.	CONFIGURATION SETTINGS FOR ROW ORGANIZED TABLES.....	12
3.	TERADATA BTEQ SUPPORT IN DCW	13
1.	CONVERT THE CODE	13
2.	EXECUTING THE SHELL FILE AND VERIFICATION	15

Database Conversion Workbench (DCW) Teradata Support

1. Introduction

Download DCW from – [Click here](#)

Link to DCW community to get tutorials and samples – [Click here](#)

DCW install on MacOS

On macOS, DCW currently has limitations on how it is invoked. Following instructions will help you to run DCW successfully on macOS -

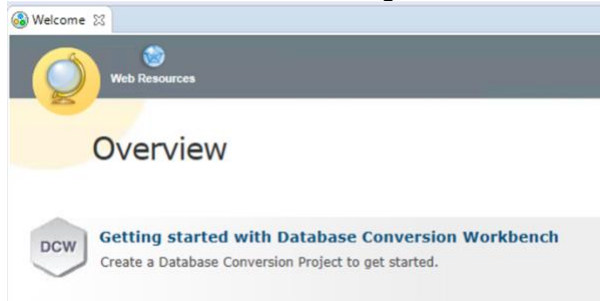
1. Check if the macOS system has Oracle JDK installed.
2. If Oracle JDK is not installed, you will need to install it. Download and install Oracle JDK from <https://www.oracle.com/java/technologies/javase-downloads.html>
3. If Oracle JDK is already installed, make sure it is greater than version 8 and up to version 14.
4. Open a terminal window, navigate to the unzipped folder and launch DCW application by running `./dcw`
5. If you receive system error about "'dcw' cannot be opened because..." then go to System Preferences, Security & Privacy all allow dcw to run as an app.
6. In the Terminal window again execute `./dcw` or `./db_harmony_profiler.sh` according to the product you have selected to launch DCW application

Mac install related known Issue: Top-level menu options of DCW are disabled

Workaround: If top-level menu options are disabled, shift focus on a different window and come back to DCW. If you receive an alert prompt saying "The DCW executable launcher was unable...." then press 'OK' and In the Terminal window again execute `./dcw` and run the application.

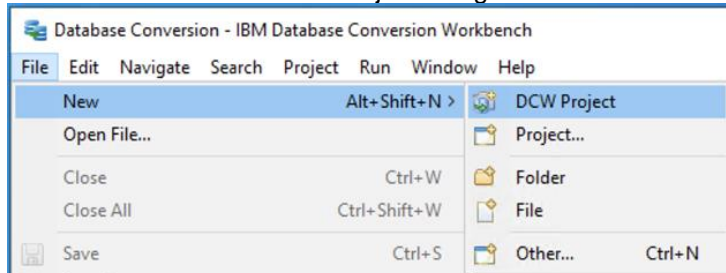
2. Table & View Support in DCW

Launch DCW and select “Getting started with Database Conversion Workbench” option

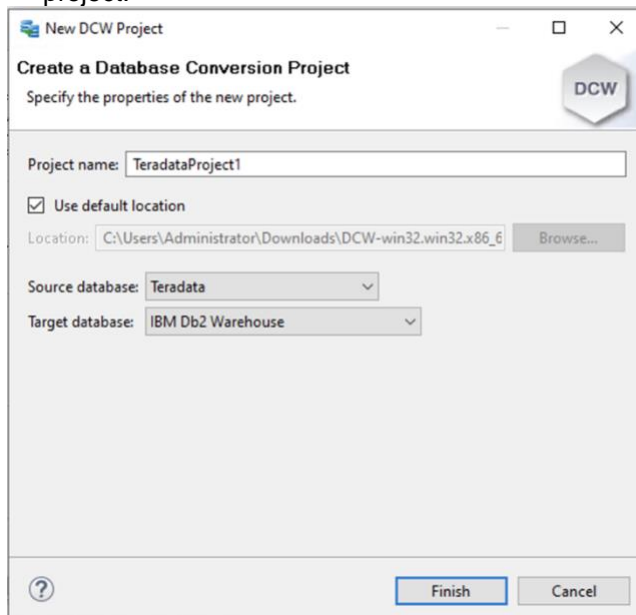


1. Create Teradata Project

- Create a new Teradata Project using “File -> New -> DCW Project”

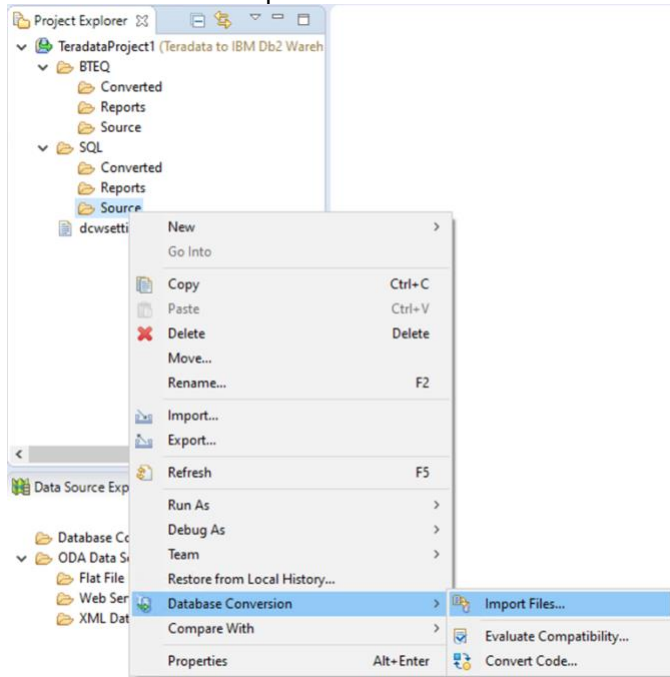


- Provide the Project Name and select “Source database” as **Teradata** and click “Finish” to create the project.

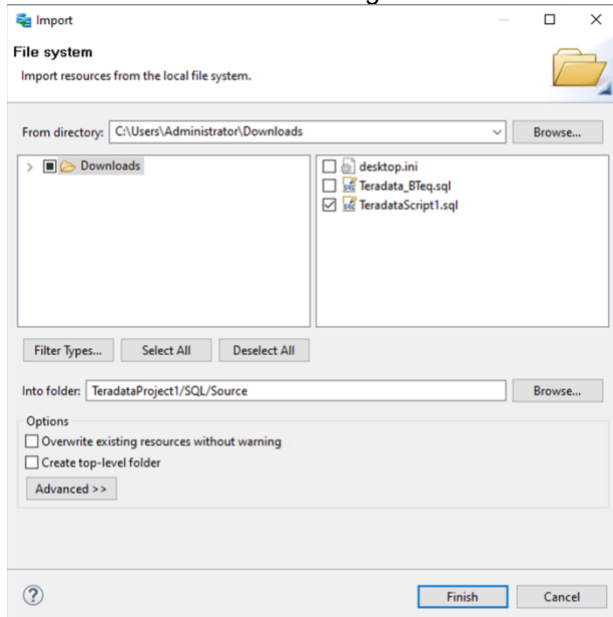


2. Import SQL File

- Expand the Project in Project Explorer and click on SQL->Source folder, then select “Database Conversion - > Import Files...”



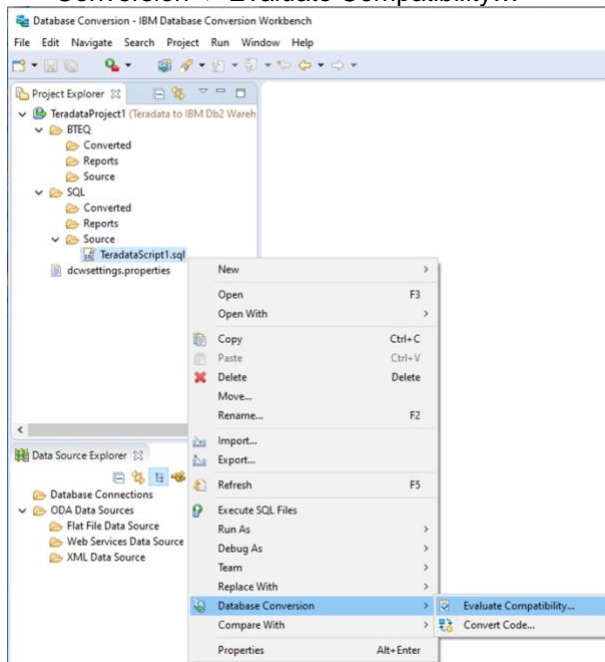
- Select the folder name using the “Browse...” button and choose the file name from the right box



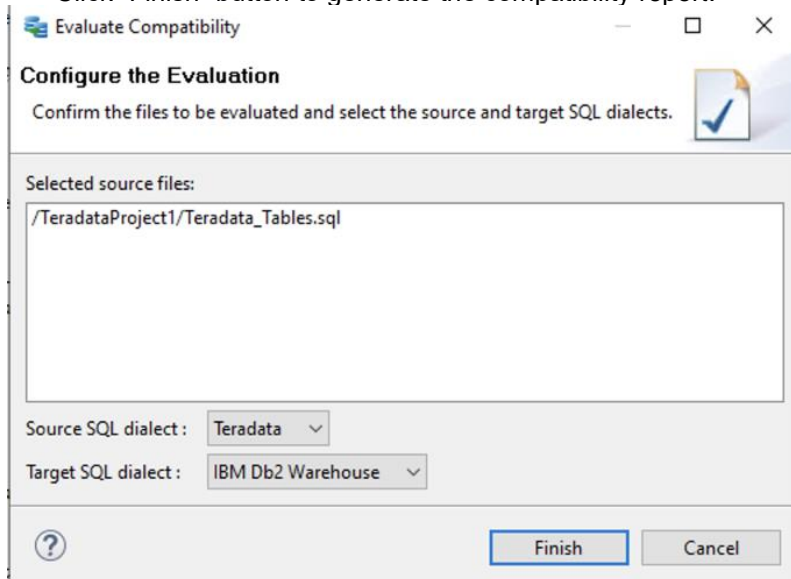
- And click “Finish” button to import the SQL file.

3. Evaluate Compatibility

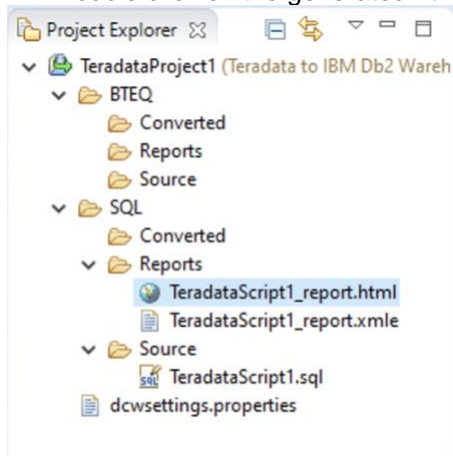
- Right click on the imported SQL file under the Project name in Project Explorer and select “Database Conversion -> Evaluate Compatibility...”



- Click “Finish” button to generate the compatibility report.



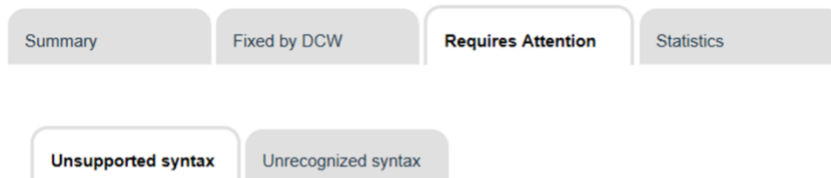
- Double click on the generated html file in the Reports folder to open the report -



- It has multiple tabs to show the details and guide on the incompatibilities requiring manual cleanup.



- You can review incompatibilities listed under “Requires Attention” tab – “Unsupported syntax” and “Unrecognized syntax” tabs and fix these issues in the sql file before re-running the compatibility report again.



Manual conversion needed

5 features are not supported by the target database and will require to be manually converted.

1 high, 2 medium and 2 low effort feature needs modification.

Click on the items below to review all occurrences and the suggested solution. Then perform the required changes in the source code.

- You can also get effort estimations to plan the migration project by looking at the “Effort Summary” table under the “Summary” tab. It lists all the incompatibilities by complexity and frequency.

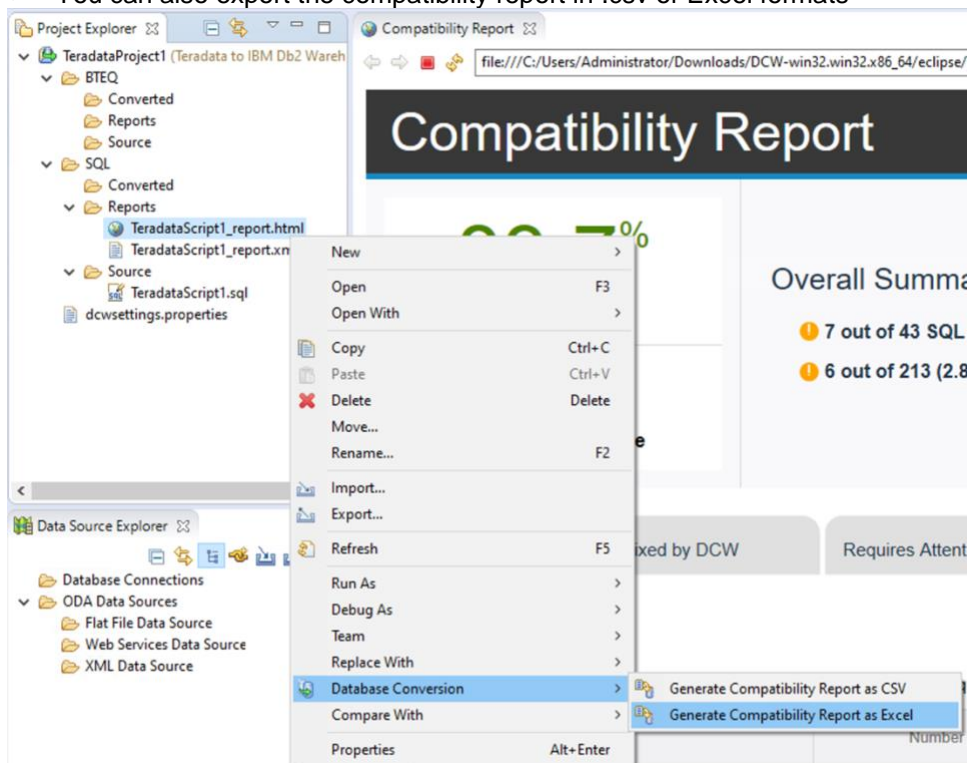


Effort Summary

*The Number of unique issues column shows the count of feature with which require similar effort of Unsupported Syntax .

Effort	Number of unique issues	Frequency
HIGH	1	1
MEDIUM	2	2
LOW	2	3
UNRECOGNIZED SYNTAX	2	2

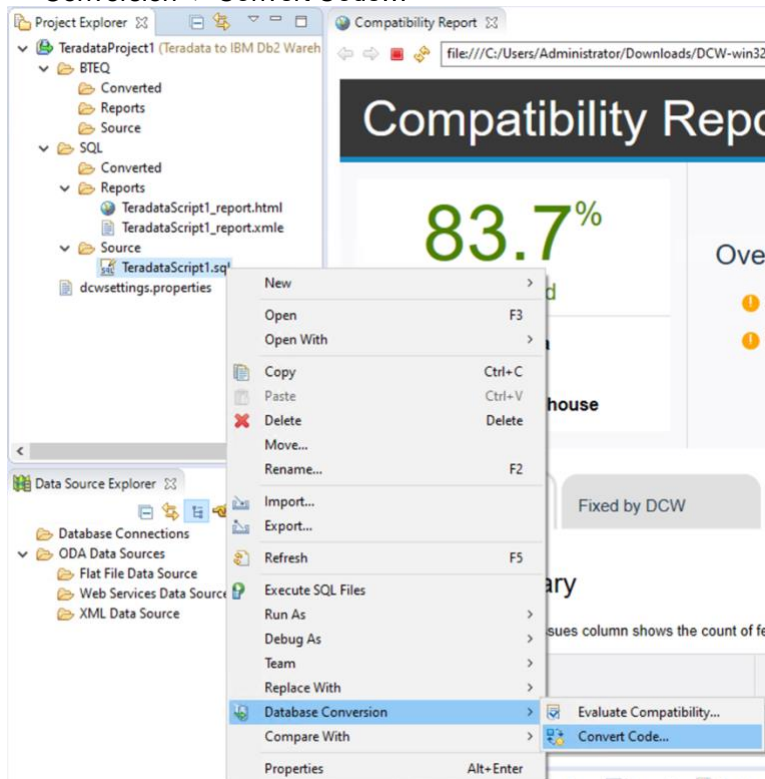
- You can also export the compatibility report in .csv or Excel formats –



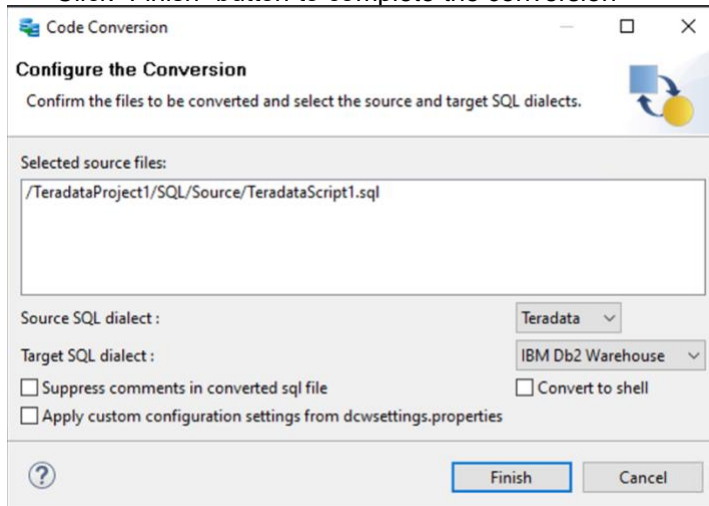
- Note:
 - Generate Evaluation report as CSV/Excel feature is available only in Full Version
 - Generate Evaluation report as CSV/Excel feature is not supported as directory level

4. Convert the Code

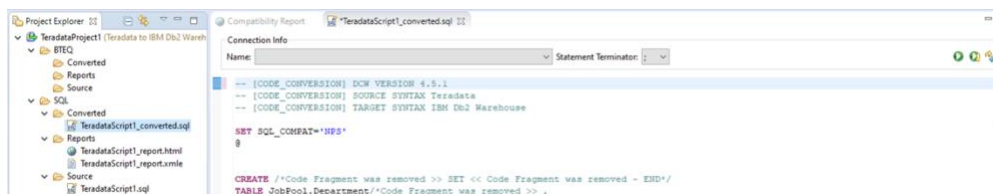
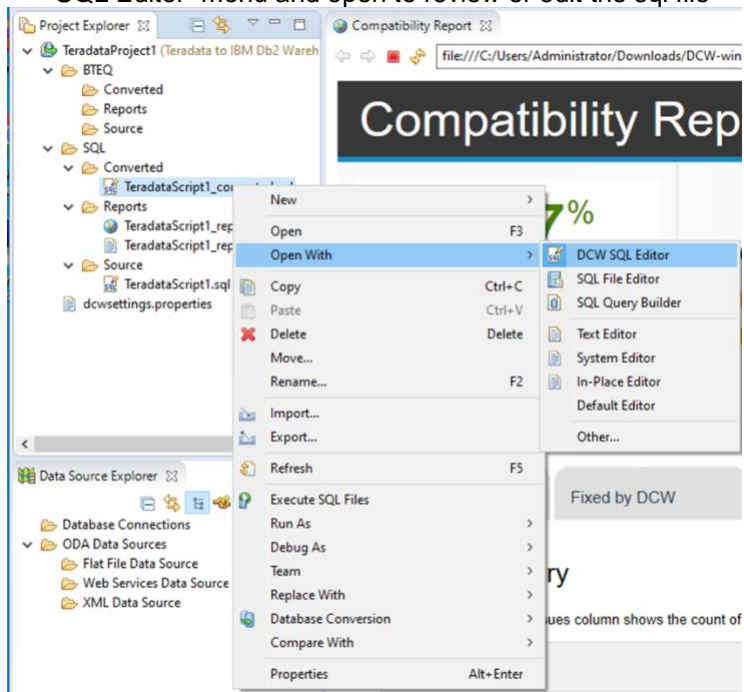
- Right click on the imported SQL file under the Project name in Project Explorer and select “Database Conversion -> Convert Code...”



- Click “Finish” button to complete the conversion

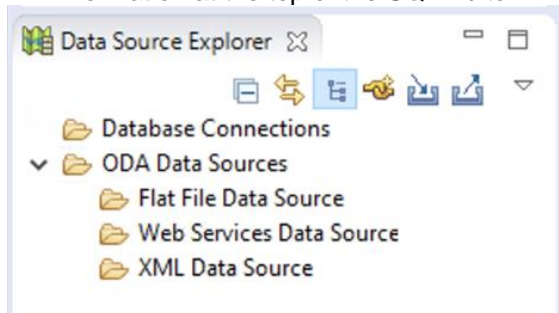


- Converted SQL is available in the project with _converted suffix, you can select “Open With -> DCW SQL Editor” menu and open to review or edit the sql file



5. Run SQL

- You can run the converted SQL code on the target Db2 Warehouse system by creating connection to the target database under Data Source Explorer and running the converted SQL using the connection information at the top of the SQL Editor.

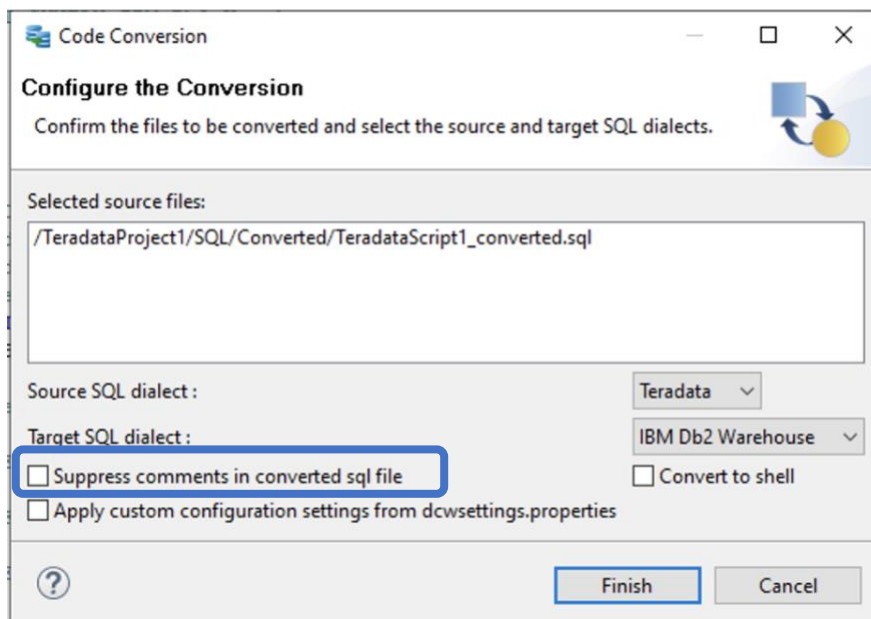


6. Suppress Comments

- This feature removes all DCW generated comments in the converted file, generating a clean and easy to read the converted file. Original user comments are still preserved in the converted file.

How to use

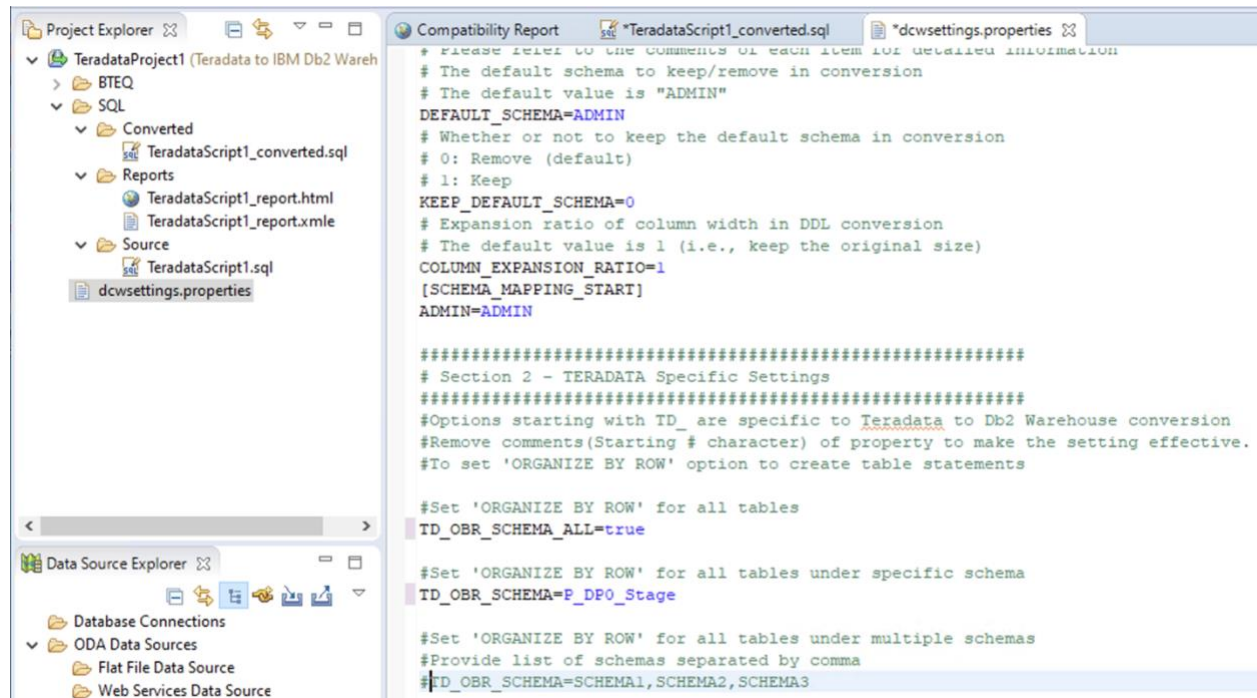
- After importing your source file, select the menu option "Database Conversion" -> "convert code" on the selected file
- Check the "suppress comment in converted sql file" check box as shown below
- Click Finish



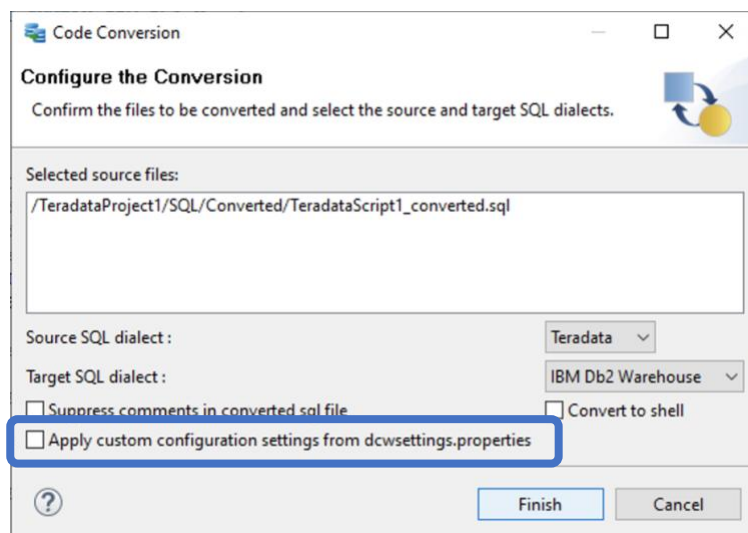
7. Configuration Settings for Row Organized Tables

The DCW config file “dcwsettings.properties” helps user set all the tables or table specific to particular schema/s with “Row Organized” distribution. This new property file will be shown in the project view now. To apply configuration file settings user will have to apply “Organize by Row” to all tables in the source file.

- Edit the “dcwsettings.properties” file as needed

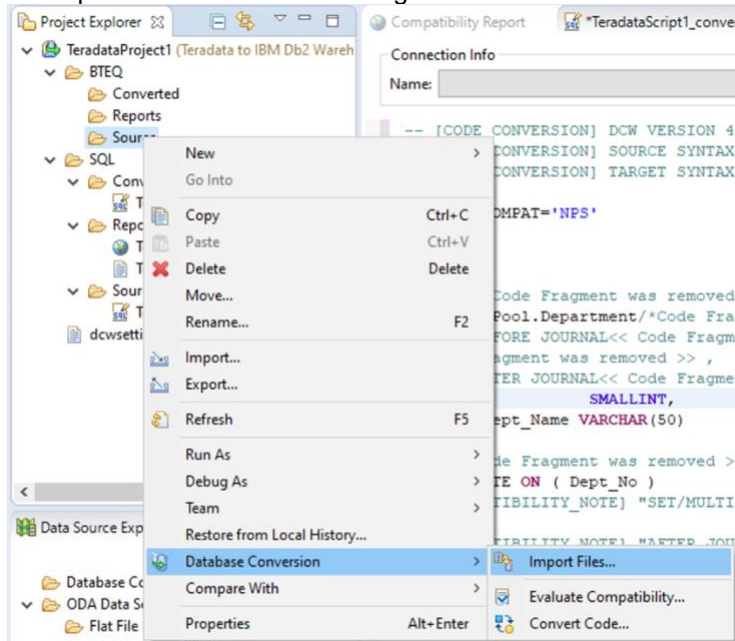


- Save the changes made
- Check the “Apply custom configuration settings” box during conversion



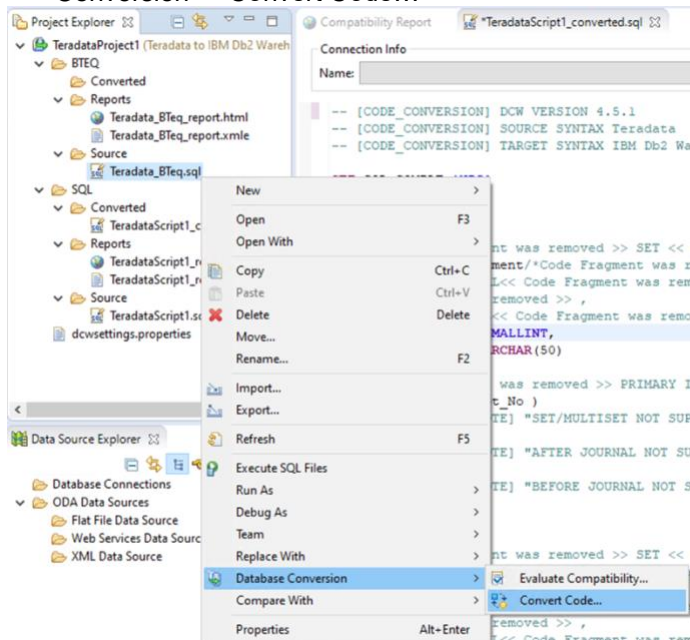
3. Teradata BTEq Support in DCW

Follow steps 1-3 as above to create a Teradata Project, Import BTEq script and Evaluate Compatibility. Select “BTEQ->Source” folder to import the BTEq script. Evaluate compatibility and fix any incompatibilities before converting code.

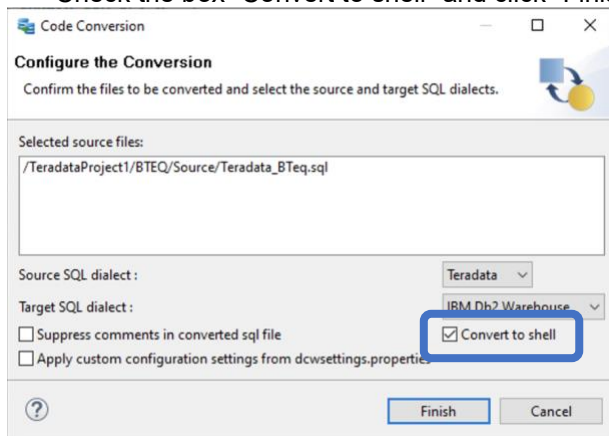


1. Convert the Code

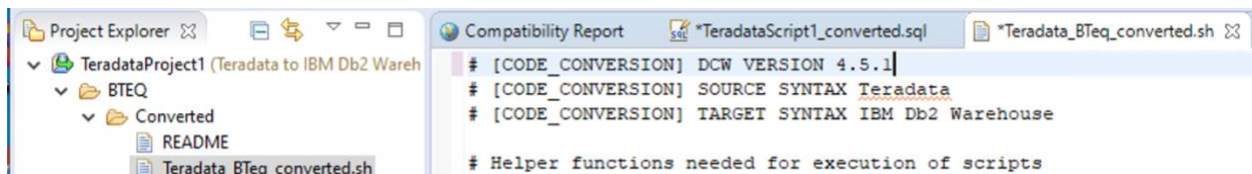
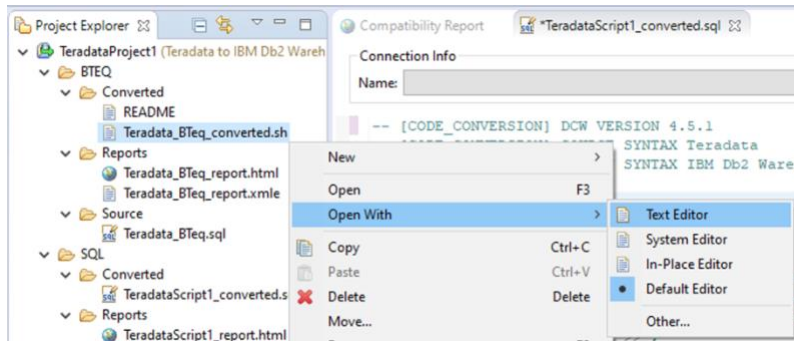
- Right click on the imported SQL file under the Project name in Project Explorer and select “Database Conversion -> Convert Code...”



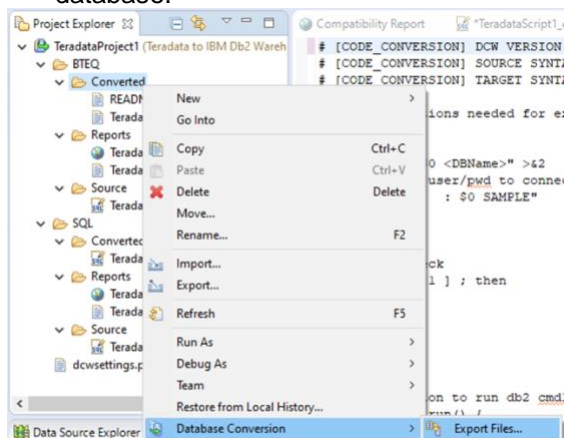
- Check the box “Convert to shell” and click “Finish” button to complete the conversion



- Once you convert the code, DCW generates a .sh file.
- You can view and edit this file using “Open With -> Text Editor” option.



- You can then export the generated .sh file to a desired folder and can deploy it on the target database.



BTEQ / SQL files can be converted to Db2 compatible SQL files, which can be tested within DCW by selecting the target database and executing against it. BTEQ / SQL files can also be converted to shell files. This can be done by selecting option “Convert to shell” in Convert wizard during conversion process. When converted to shell, each of SQL statement is wrapped in a shell function that will be executed by the db2 command line processor. The error returned is checked by trap_errors() function and proceeds only if there are no errors. For transaction statements, all the SQL are enclosed in one block and executed in DB2 equivalent BEGIN/END transaction block. This transaction block is executed in a single shell function call.

The generated shell files need to be exported and executed on a Linux system where db2 command line processor needs to be accessible. A README file is generated in the Converted folder which details the requirements and instructions on how to execute a shell file in a DB2 Warehouse Environment.

- You will have to log system hosting Db2 warehouse environment and open a terminal window.

```
[bluadmin@dcwsystem1 - Db2wh 9]$ unzip bteq-shell.zip
Archive:  bteq-shell.zip
  inflating: TeradataConversionProject/Teradata_Bteq_converted.sh
[bluadmin@dcwsystem1 - Db2wh 9]$ ls -lr TeradataConversionProject/
total 4
-rw-rw-r-- 1 bluadmin bluadmin 2904 Jul 29 19:47 Teradata_Bteq_converted.sh
[bluadmin@dcwsystem1 - Db2wh 9]$
```

2. Executing the shell file and verification

- Execute:

```
sh Teradata_Bteq_converted.sh
```

```
Downloads — bluadmin@dcwsystem1:~/sjorapur/nov03/TeradataProject1/BTEQ/Converted — ssh root@dcws
[bluadmin@dcwsystem1 - Db2wh Converted]$ sh Teradata_Bteq_converted.sh

Usage: Teradata_Bteq_converted.sh <DBName>
Uses OS user/pwd to connect to DB
Example   : Teradata_Bteq_converted.sh SAMPLE

[bluadmin@dcwsystem1 - Db2wh Converted]$
```

- Note that you need to add in Database name as an argument to the shell file. It will connect to the given database and execute the commands. This will use logged in user's credentials to connect to database. However, if credentials differ, then edit the shell file and set up the username/password. Make sure to uncomment the statement as follows -

```
#Start
```

```
db2 connect to $DBNAME USER <UserName> using <Password>
```

```
#db2 connect to $DBNAME
```

- Example when BLUDB is name of the database
sh Teradata_Bteq_converted.sh BLUDB

```
[bluadmin@dcwsystem1 - Db2wh Converted]$ sh Teradata_Bteq_converted.sh BLUDB

Database Connection Information

Database server      = DB2/LINUX8664 11.5.4.0
SQL authorization ID = BLUADMIN
Local database alias = BLUDB

Connection succeeded...

In trap errors
DB20000I The SQL command completed successfully.

SQL succeeded

In trap errors
DB20000I The SQL command completed successfully.

SQL succeeded
```

- Verifying by issuing the following command in the terminal window to verify the table was created successfully with data in it.

```
[bluadmin@dcwsystem1 - Db2wh TeradataConversionProject]$ dbsql -u bluadmin -pw bluadmin -d bludb -c "SELECT * FROM WORLD.COUNTRIES"
ID | NAME
-----+-----
IND | India
(1 row)
```



© Copyright IBM Corporation 2020 All Rights Reserved.

IBM Canada
8200 Warden Avenue
Markham, ON
L6G 1C7
Canada

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at ibm.com/legal/copytrade.shtml.

Oracle and JDBC are trademarks or registered trademarks of Oracle Corporation in the United States, other countries, or both.

Linux is trademark or registered trademark of Linux Torvald in the United States, other countries, or both.

Other company, product and service names may be trademarks or service marks of others.

References in this publication to IBM products and services do not imply that IBM intends to make them available in all countries in which IBM operates.

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

IBM products are warranted according to the terms and conditions of the agreements (e.g. IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided.