Quest™

Toad® DevOps Toolkit™

Power Shell Demonstration

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.

Attn: LEGAL Dept

4 Polaris Way

Aliso Viejo, CA 92656

Refer to our Web site (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal .

Trademarks

Toad® DevOps Toolkit™

Power Shell DemonstrationToad® DevOps Toolkit™
Updated – September 25, 2017
Version – 1.0

Contents

Purpose

This example will demonstrate how PowerShell can be used to call TDT commands and return results.

The examples provided show how to:

1. Check Database Connections.
2. Run Code Analysis.
3. Run Execute Script.
4. Run Compare Databases.
5. Run Compare Schemas.
6. Run Compare Tables.
7. Run Unit Testing (using Code Tester).
8. Run Debug.

Requirements

PowerShell

This demo utilizes PowerShell Scripts to make TDT calls. The scripts were written in PowerShell version 5.0. The process will check that you are running at least v3.0 and will fail if you are running a lower version. To install PowerShell version 5.0, download Windows Management Framework 5.0, which can be located here: PowerShell 5.0 Download

Oracle

Two separate databases are required – Source and Target. They need to be completely separate databases. Also, to install the DEMO schema on these databases, the user must have SYSDBA privileges to run the install process.

Code Tester

Code Tester is not required to run this demo, however, to run Unit Testing, a Code Tester Repository must exist on the database and must be at least version 3.2.0.2. If the repository is not found or not later than 3.2.0.2, then the Unit Testing module will be skipped.

Configuration

This process requires no programming or PowerShell experience. It uses one main configuration file (**Config.ps1**) that controls the database login information, directories, files, input, output, modules, etc. It contains all the major parameters that someone might want to change to run the different modules. For example, the user may *not* want Code Analysis to output a JSON report – in this case, they would edit the **Config.ps1** file, navigate to the Code Analysis section and set the parameter to False. (`$CodeAnalysisReportFormatsIncludeJSON = $FALSE`). Additionally, each module has a run flag setting, which when set to $FALSE, will skip that module from running.

Modules

An example is provided for each TDT module. A description of what each module does is explained in more detail below.

1. **Validation Checks**
    a. Verifies TDT is installed.
    b. Verifies TDT can be started and run.
    c. Verifies TDT can connect to the source and target databases.
    d. Verifies TDT can be stopped.

2. **Code Analysis**
    a. Processes both Files and DB Objects
    b. Saves and Loads XML Properties Files
    c. Executes Code Analysis
    d. Creates Output reports

3. **Execute Script**
    a. Add/Delete Script Files
    b. Executes Script Files
    c. Creates Output reports

4. **Compare Databases**
    a. Loads/Saves properties to/from snapshot files
    b. Loads properties from configuration file
    c. Exclude Objects
    d. Execute Compare
    e. Creates Output reports and sync scripts

5. **Schema Compare**
    a. Loads/Saves properties to/from snapshot files
    b. Loads properties from configuration file
    c. Exclude Objects
    d. Execute Compare
    e. Creates Output reports and sync scripts

6. **Table Compare**
    a. Loads/Saves properties to/from snapshot files
    b. Loads properties from configuration file
    c. Execute Compare
    d. Creates Output reports and sync scripts

7. **Code Tester Checks**
    a. Checks Repository Version and Status on Source Database
    b. Checks Repository Version and Status on Target Database
    c. Sets Unit Testing Run Flag based on Version and Status checks.

8. **Unit Testing**
    a. Loads properties from configuration file
    b. Exports Unit Tests and Suites to XML files (from Source DB)
    c. Import Unit Tests and Suites from XML files (to Target DB)
    d. Runs Unit Tests and Suites (in Target DB)

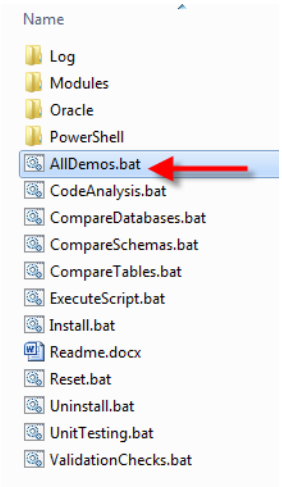
9. **Debug**
    a. Displays TDT Debug and Exception Log Filenames
    b. Exports Debug and Exception Log Information

10. **Display Enums**
    a. Displays Enums that can be used (from TDT_enums.ps1)

Run Demo

For individual module runs, double click on {Module}.bat name, or 'AllDemos.bat' to run all modules. All Demos is used in the example below.

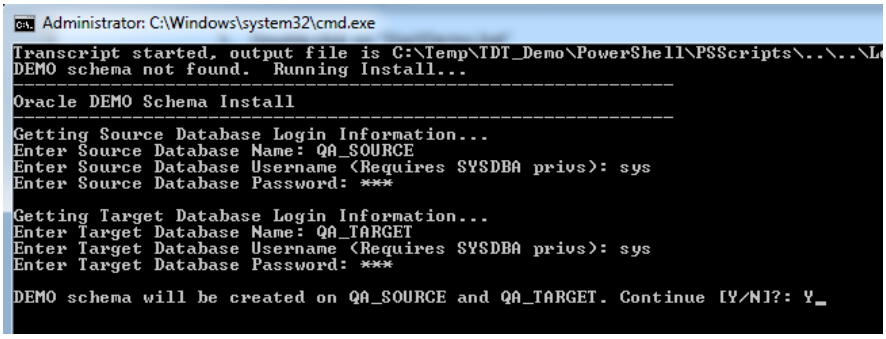
1. Double click on 'AllDemos.bat' (or the .bat file of the module you want to run)



2. If this is the first time running, Install.ps1 will be run and require input for both a Source and Target database login information (see Requirements section for more information).

   Notes:

   1. SYSDBA privileges are required for installing the DEMO schema.
   2. DEMO schema is installed on both the Source and Target Databases.
   3. Database information is saved to Config.ps1.
   4. Log of Install will be saved to OracleDemoSchemaInstall_Log.txt



Output

Logs

A transcript (log) of the entire run will be saved to the **Log** Directory, and will also automatically open after the process has finished.

1. OracleDemoSchemaInstall_Log.txt – generated after doing an initial install of the Demo schema.
2. {YYYYMMDDHHMMSS}_Log.txt – generated after running a module (or all modules for AllDemos).
3. OracleDemoSchemaUninstall_Log.txt – generated after doing an uninstall of the Demo schema.

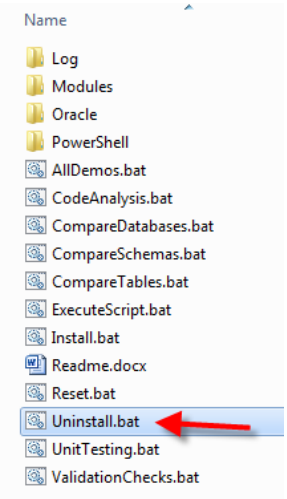
Modules

If a module produces any output files, they will be saved to the **Modules\{Module Name}\Output** Directory. There are some pre-canned input files that are used during the process, and they will be located in the \Input directories.

Uninstall

To remove the DEMO schemas from the Source and Target Databases, you can either drop the Demo users on each database manually, or run Uninstall.bat.

Note: Database information is removed from Config.ps1 and the user will be prompted to reinstall the Demo schema on the next run.



Reset

Reset is a quick way to remove any log or output files that were created from a previous run.

Note: Database information is *not* removed from Config.ps1 and the user will not be prompted to reinstall the Demo schema on the next run.