**Netezza DDL Converter Utility**

The Netezza DDL converter utility is tool to convert Netezza database schema DDL to SQL equivalent. The highlight of this utility is that it can convert Netezza MPP DDL to SQL MPP or SMP equivalent, which means it supports the conversion to Azure SQL DW, Azure SQL DB, Azure SQL MI and SQL Server on-premises.

This is a Windows console-based utility and currently supports conversion of table schema, defaults and constraints. Following features are currently supported by the utility:

* **Azure Conversion**
  + Azure SQL DW
    - Standard Table Conversion
    - External Table Creation & Conversion
  + Azure SQL DB, Azure SQL MI
    - Standard Table Conversion
    - Defaults and Constraints Conversion
* **On Premises**
  + SQL Server
    - Standard Table Conversion
    - Defaults and Constraints Conversion

**Prerequisites**

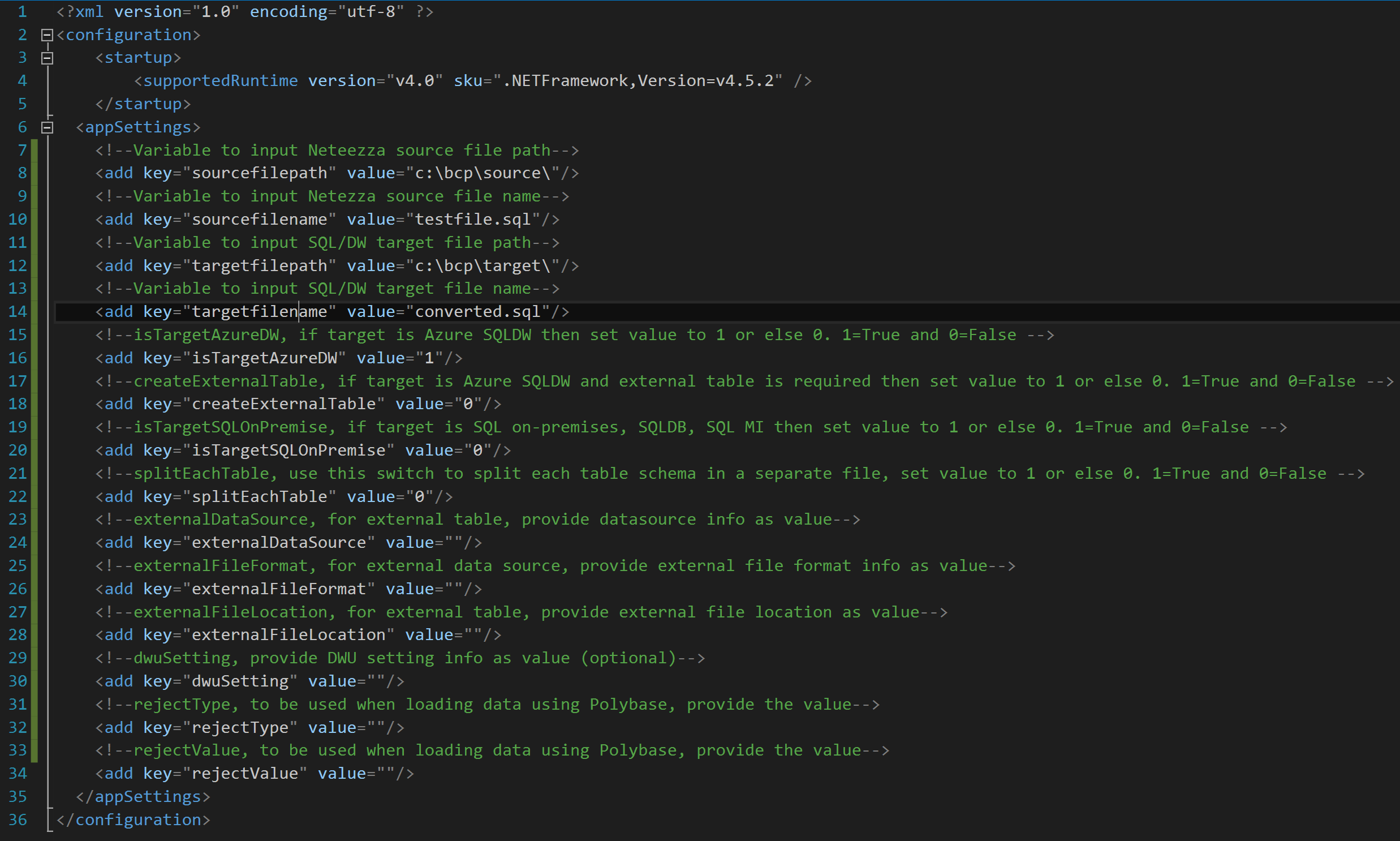
Following are the prerequisites to run this utility:

* Netezza DDL is extracted using the Netezza Tools like nzsql or Agility Workbench. DDL is extracted into a single file. This should just be the Create Tables with defaults and constraints. Views, Stored Procedures and other Netezza objects are considered out of scope for hit tool at this time.
* Azure SQL DW is configured for Polybase
* Data Source for Polybase has been created
* File Format for Polybase has been created

**Components of Utility**

This utility has two components

1. **Netezza Converter.exe.config** – This is a XML config file which stores all configuration related to conversion. One must edit this file to update desired conversion configurations.



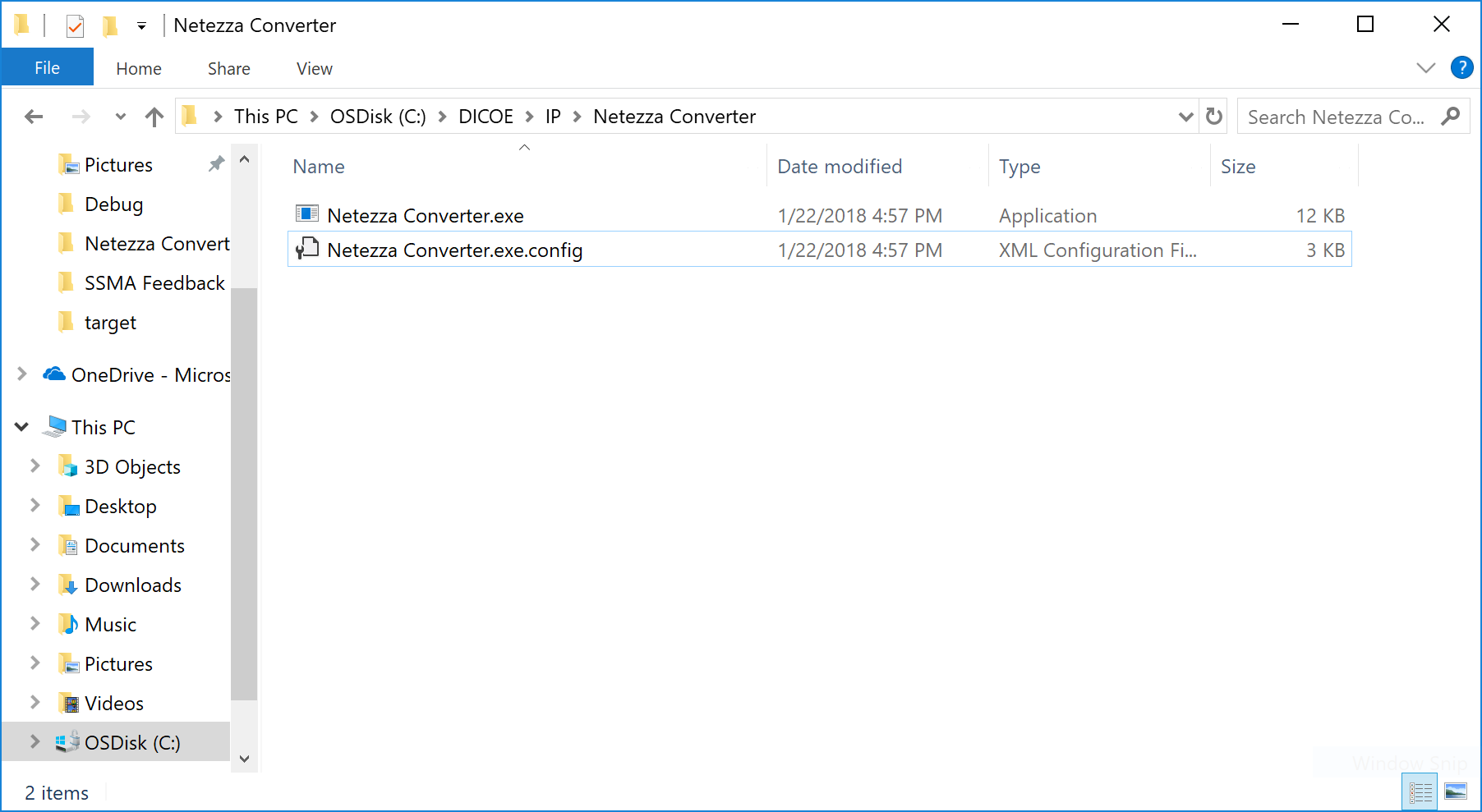
Here are the details to configurations settings to update before running the utility.

* + **sourcefilepath** – Update the file path of Netezza DDL source file
  + **sourcefilename** – Update the name of Netezza DDL source file
  + **targetfilepath** – Update the file path for converted DDL
  + **targetfilename** – Update the name of converted DDL file “filename.sql”
  + **isTargetAzureDW** – Update the value to 1 if conversion target is Azure SQL DW
  + **createExternalTable** – Update the value to 1 if target is Azure SQL DW and you also want to create external tables
  + **isTargetSQLOnPremise** – Update the value to 1 if conversion target is Azure DB, SQL MI or SQL Server on-premises
  + **splitEachTable** – Update the value to 1 if converted DDL should create one DDL file for each table. In case if Azure SQL DW with external table switch is on, this setting generates a single file including external table
  + **externalDataSource** – Update the name of Polybase data source created in Azure SQL DW
  + **externalFileFormat** – Update the name of Polybase file format created in Azure SQL DW
  + **externalFileLocation** – Update the location of WASB where data files are stored. This is the directory structure. It is assumed that each table will have its own location based on table name
  + **rejectType** -Update rejectType to use when loading data using Polybase
  + **rejectValue** -Update rejectValue to use when loading data using Polybase

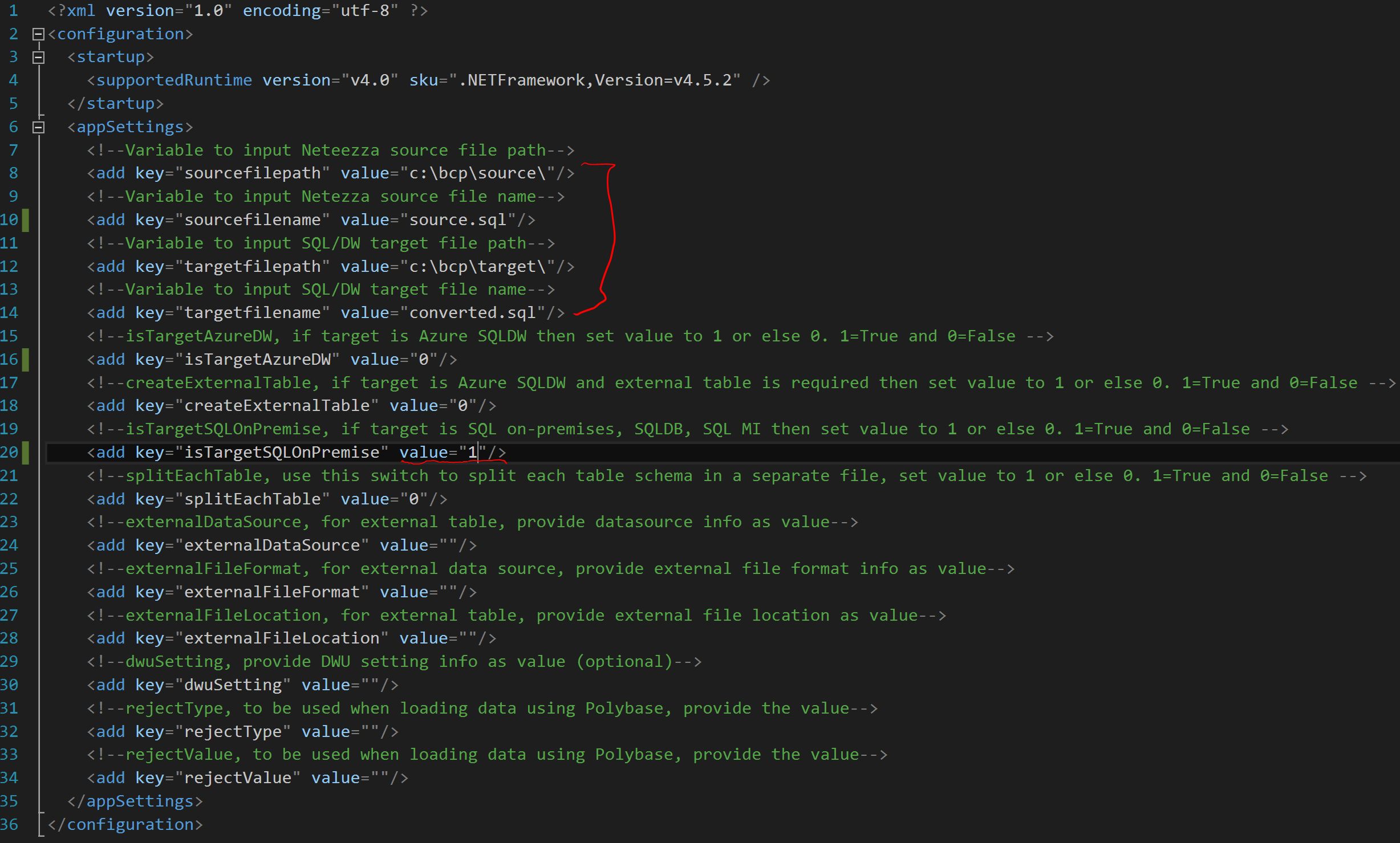
1. **NetezzaConverter.exe** – This is an executable file and should be invoked from a command prompt. Once the XML configuration is updated, we must simply execute this exe from command prompt and the converted DDL is generated in the target folder.

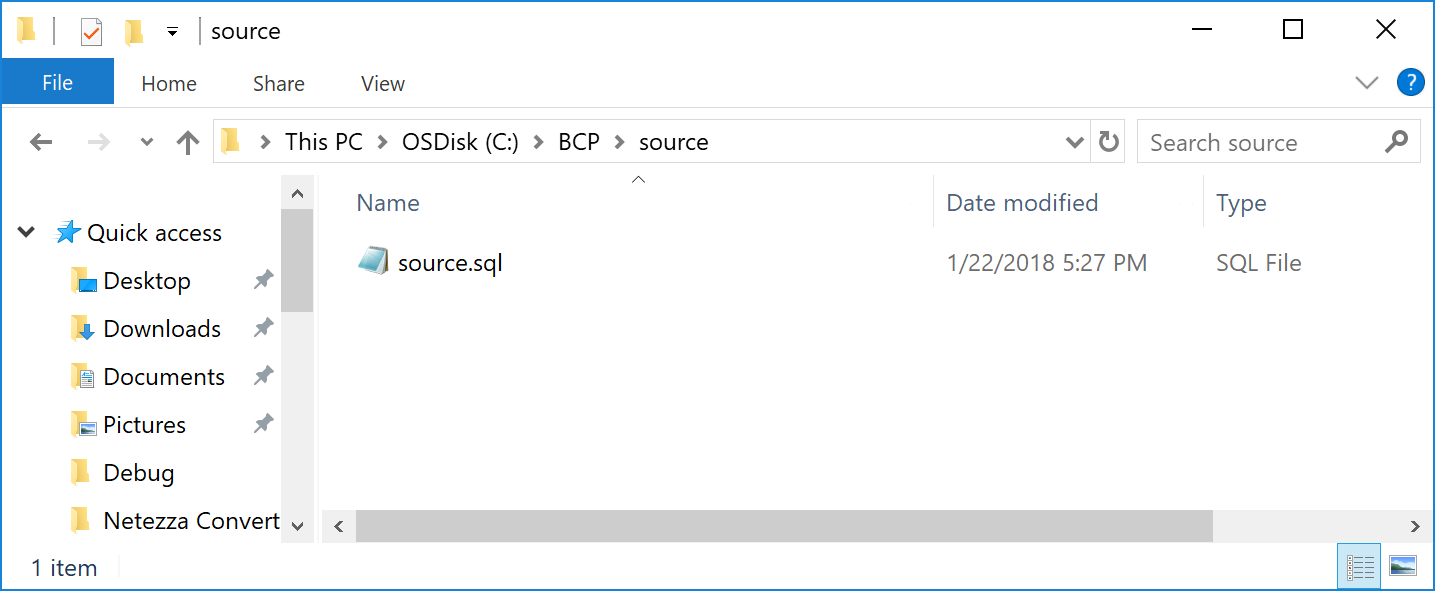
**Sample Run**

1. Download and switch to the folder having Netezza Converter exe and XML config file

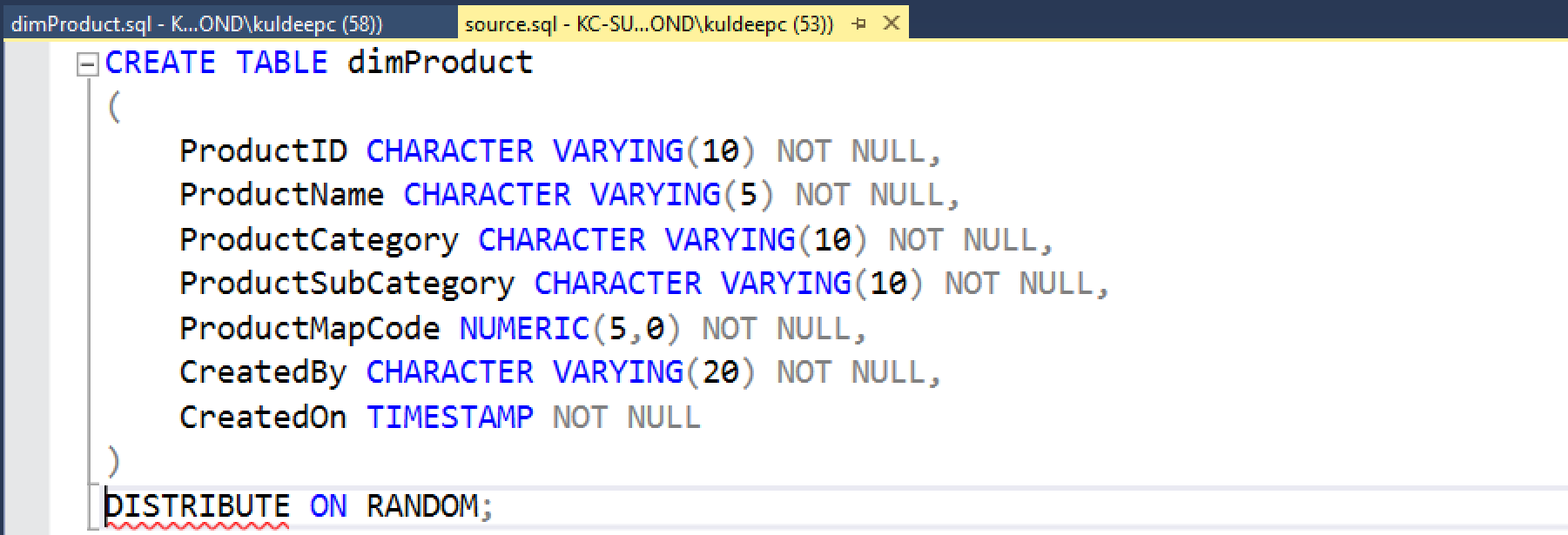


1. Edit and update the XML file with desired configuration, in the example we are converting a Netezza DDL to SQL on-premises with default and constraints conversion

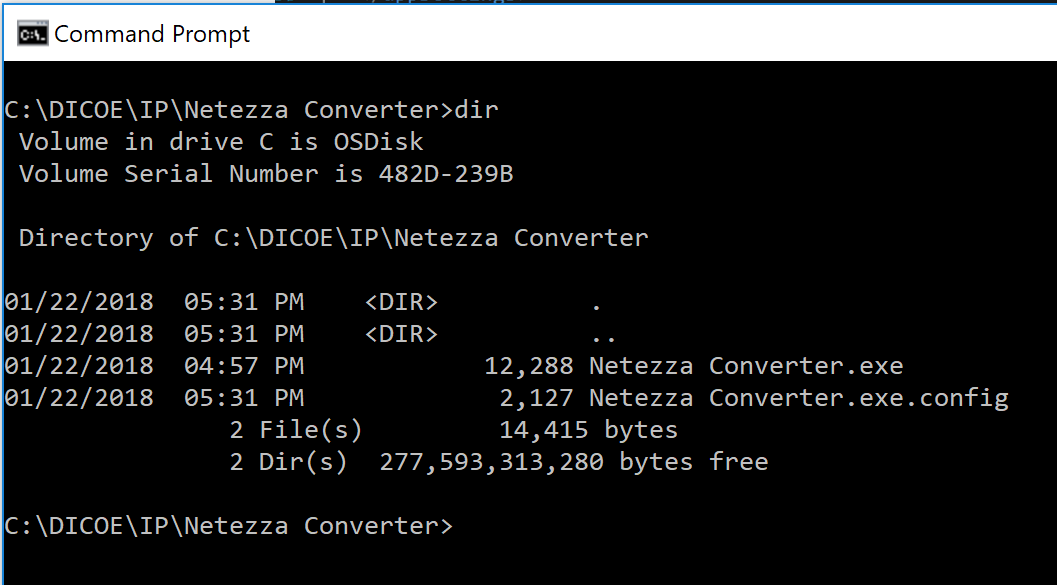




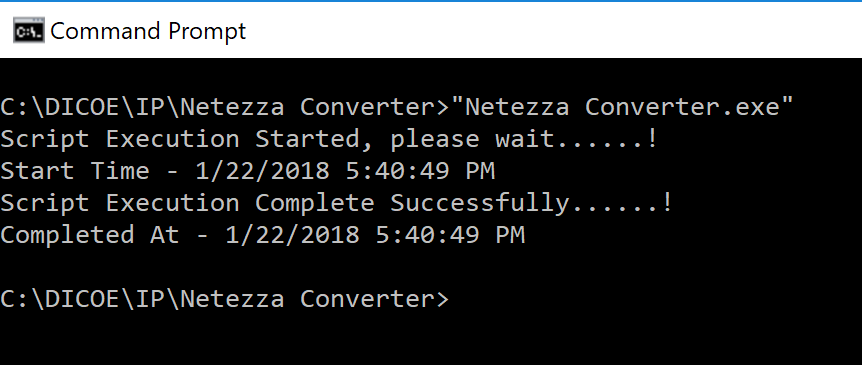
1. This is how Netezza source DDL with defaults looks like



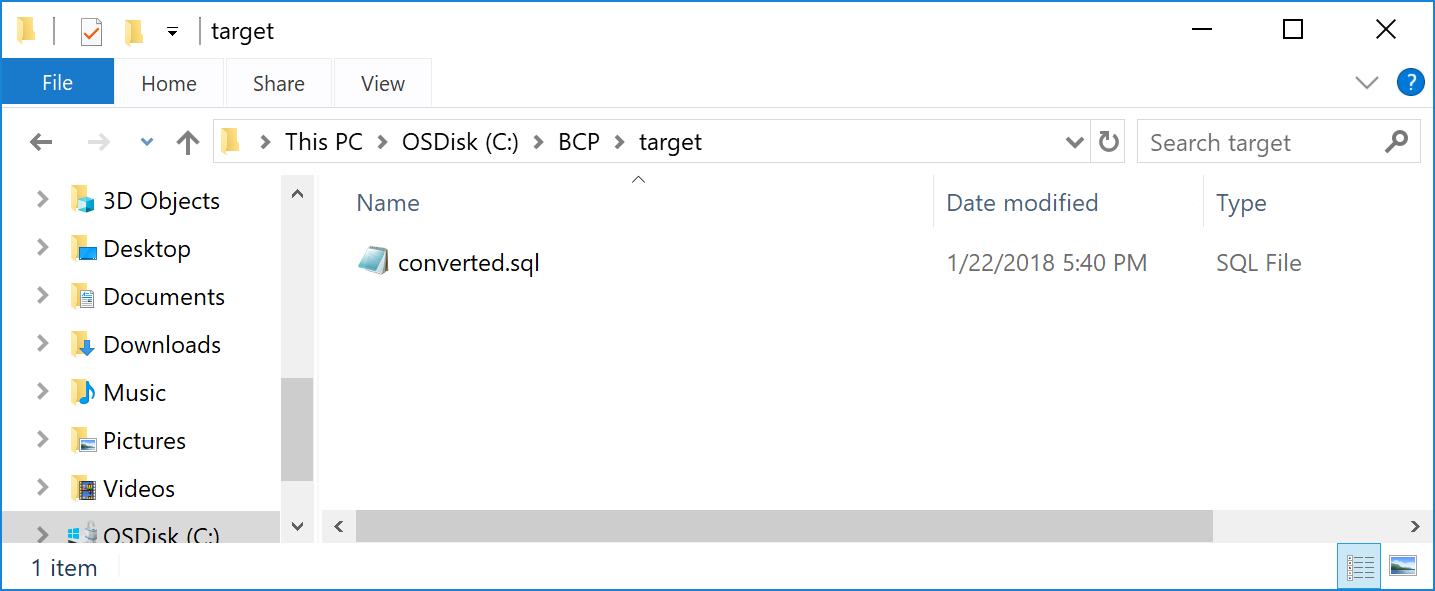
1. Open the command prompt and switch to the directory having “Netezza Converter.exe”



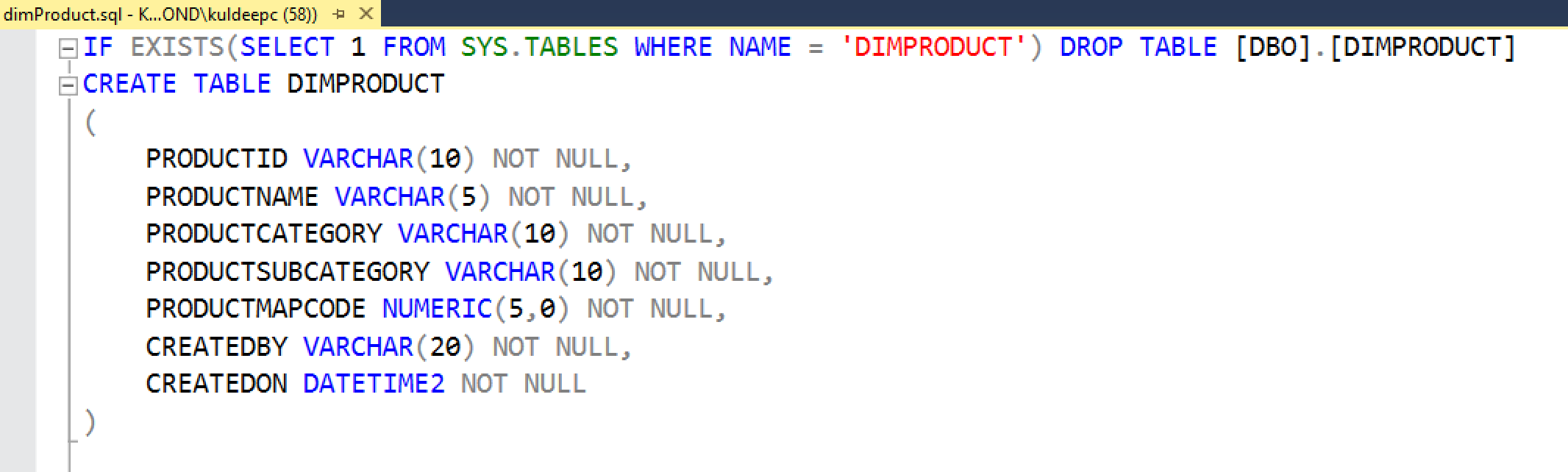
1. Execute the “Netezza Converter.exe”



1. Look into target folder and see the converted script is generated



1. Open the script in SSMA and validate that conversion is successful



Similarly, we can use this conversion utility to generate target schema based on different settings, for an example –

* Converting Schema to Azure SQL DW with or without external tables
* Split tables in separate DDL files