

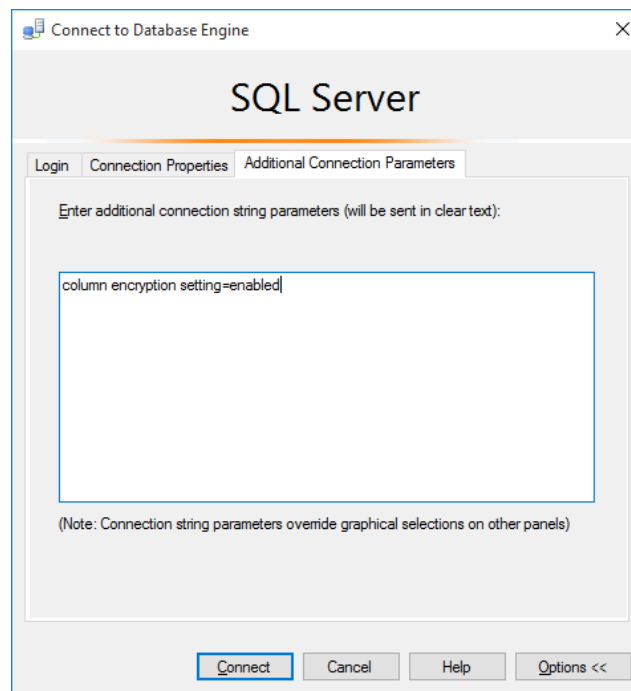
## Always Encrypted README

The AdventureWorks database uses the [Always Encrypted](#) feature to protect sensitive information stored in the `SSN` and `CreditCardNumber` columns in the `Sales.CustomerPII` table. This file provides the steps to explore the capabilities of Always Encrypted and the encryption configuration in the database.

For information about Always Encrypted, please see:

- MSDN documentation: [Always Encrypted \(Database Engine\)](#)
- Always Encrypted articles on [SQL Server Security Blog](#)

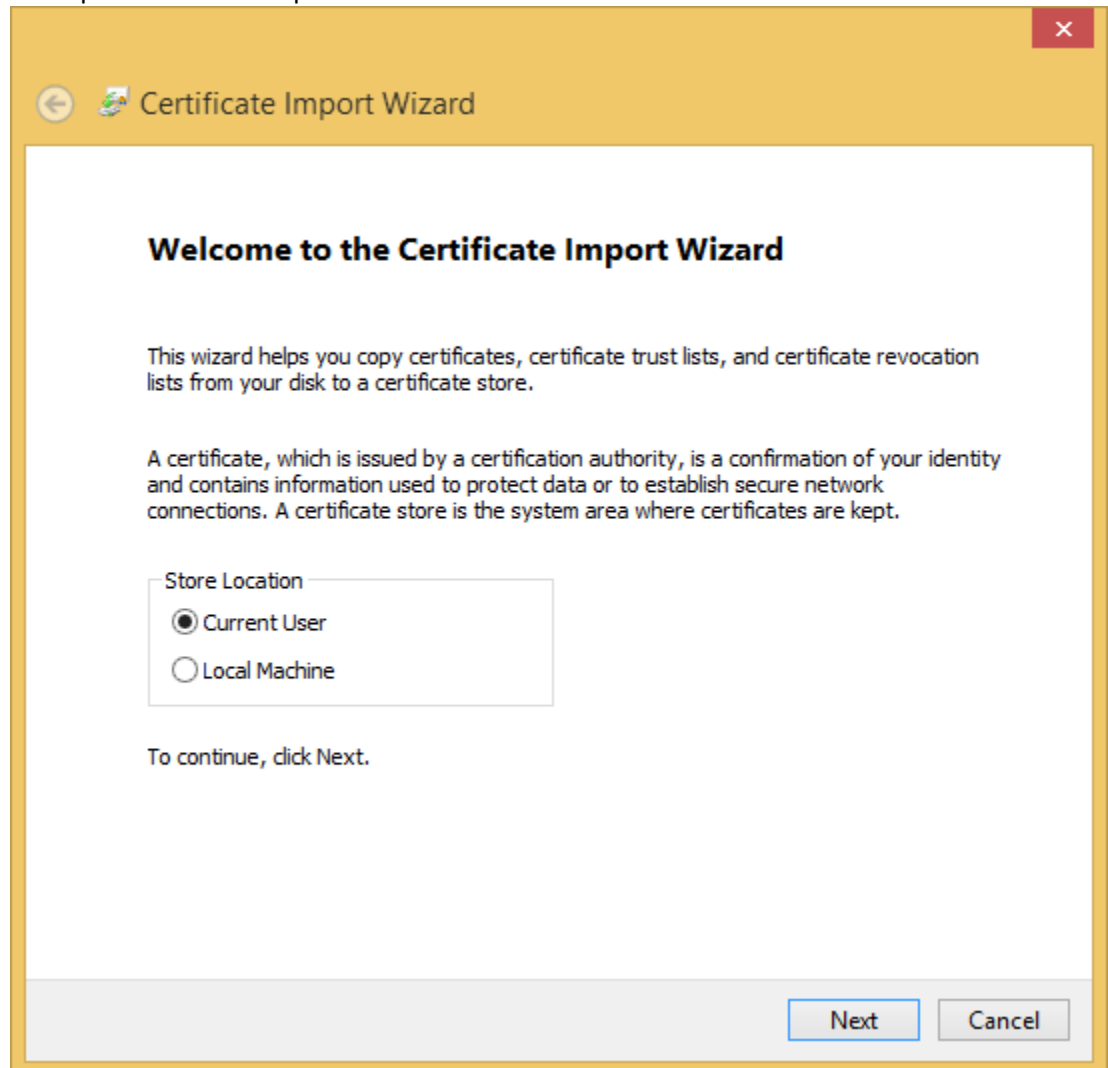
- 1) Query encrypted columns without decrypting sensitive data:
  - a) Using SSMS, connect to the database.
  - b) Using Object Explorer, navigate to the `Sales.CustomerPII` table.
  - c) Right-click on the table and choose **Select Top 1000 Rows**.
    - i) The query should return binary values, e.g.: `0x01F7753C73CA15965E314...`, in both the `SSN` and `CreditCardNumber` columns.
- 2) Query encrypted columns, attempting to decrypt sensitive data without having the valid column master key configured in your environment.
  - a) Close the database connection from step 1 and reconnect by adding the following connection string keyword/value in the **Additional Connection Parameters** of the **Connect to Database Engine** dialog: **column encryption setting=enabled**.



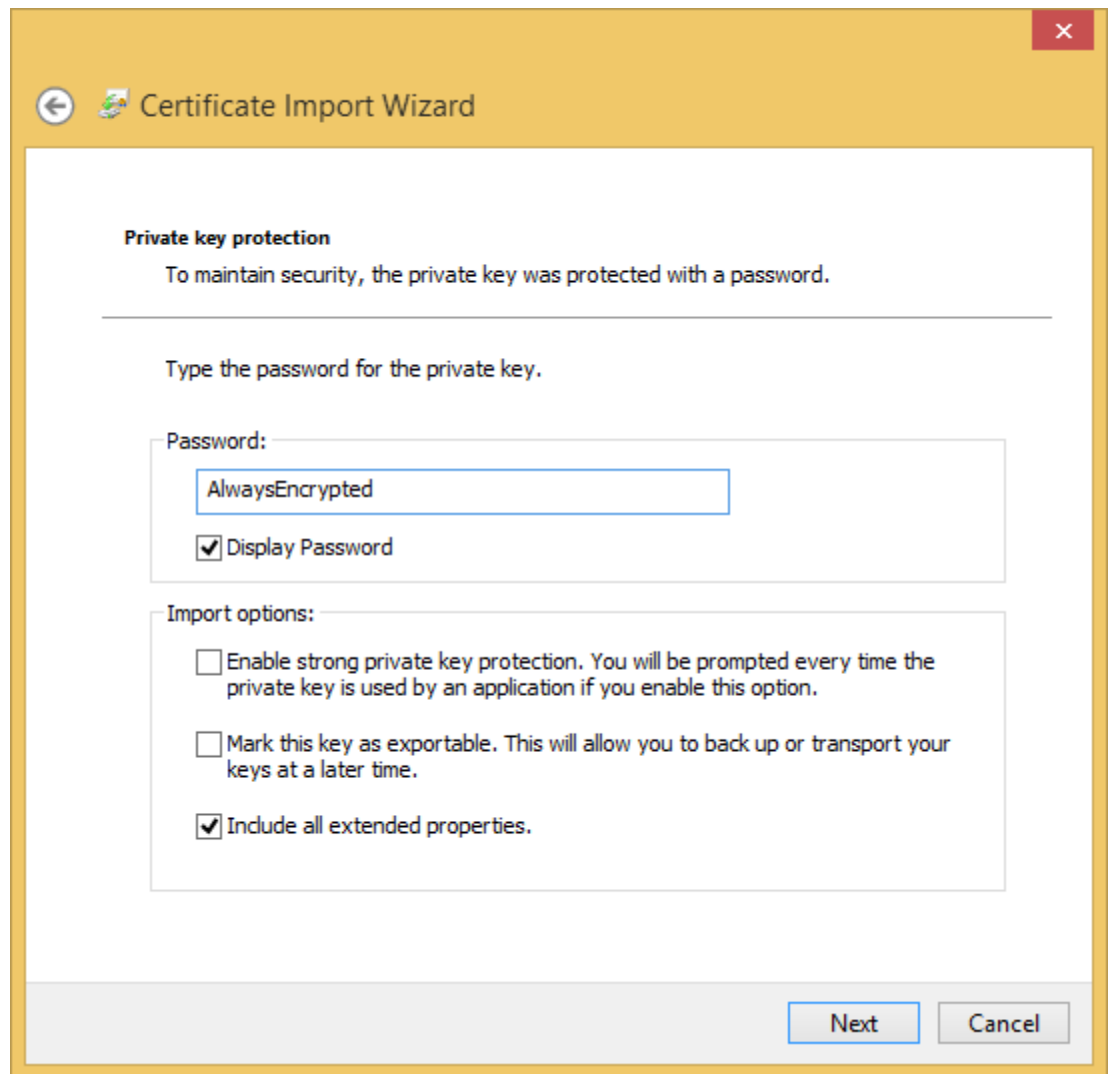
- b) Using Object Explorer, navigate to the `Sales.CustomerPII` table.
- c) Right-click on the table and choose **Select Top 1000 Rows**.

The query is expected to fail, as you try to decrypt sensitive data stored in the `SSN` and `CreditCardNumber` columns, but you do not have the column master key, protecting those columns.

- 3) Query sensitive data columns, decrypting the results using a valid column master key configured in your environment.
  - a) Import the certificate that is configured as a column master key in the database.
    - i) Using Windows explorer, navigate and double click on the `AlwaysEncryptedCMK.pfx` file. This will open Certificate Import Wizard.



- ii) Click **Next** (leave **Current User** selected for **Store Location**) and then **Next** again.
- iii) Enter the password for the certificate: **AlwaysEncrypted**.



The image shows a Windows-style dialog box titled "Certificate Import Wizard". It has a yellow header bar with a back arrow icon on the left and a close button (X) on the right. The main content area is white and contains the following elements:

- Private key protection**
  - To maintain security, the private key was protected with a password.
- Type the password for the private key.
- Password:**
  - A text input field containing the text "AlwaysEncrypted".
  - A checkbox labeled "Display Password" which is checked.
- Import options:**
  - ☐ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.
  - ☐ Mark this key as exportable. This will allow you to back up or transport your keys at a later time.
  - ☒ Include all extended properties.

At the bottom right of the dialog, there are two buttons: "Next" and "Cancel".

- iv) Click **Next** and then click **Finish**.
- b) Make sure, you are connected to the database with **column encryption setting=enabled**. (please, see step 2a).
- c) Using Object Explorer, navigate to the `Sales.CustomerPII` table.
- d) Right-click on the table and choose **Select Top 1000 Rows**.  
Since you can now access the certificate, used as a column master key in the database, the query should succeed and return plain text values stored in the `SSN` and `CreditCardNumber` columns.