Enabling Secure Sockets Layer for a Microsoft SQL Server JDBC Connection

Secure Sockets Layer (SSL) is the standard security technology for establishing an encrypted link between a web server and a browser. This use case describes how to enable SSL for a Microsoft SQL Server JDBC connection.

The following list provides a summary of the connection properties that must be used to enable SSL:

- **encrypt=** Set to *true* or *false*.
- **trustServerCertificate=** Set to *true* or *false*.
- **trustStore=** Provide the full path to the truststore including the truststore filename.
- **trustStorePassword=** Provide the password of the truststore.
- **hostNameInCertificate=** This value must match the Subject value for hostname within the certificate.

Testing With a Self-signed Certificate

When the **Encrypt** and **trustServerCertificate** properties are set to *true*, the Microsoft JDBC Driver for SQL Server will not validate the SQL Server SSL certificate. This is usually required when allowing connections in a test environment (when the SQL Server instance has only a self-signed certificate).

For example:

```
--------------------------------------------------------------------------------
jdbc:sqlserver://hostname:1433;databaseName=AdventureWorks2014;encrypt=true;
trustServerCertificate=true;trustStore=C:\Program Files\Java\jdk1.7.0_79\jre\
lib\security\cacerts;trustStorePassword=changeit
--------------------------------------------------------------------------------
```

Testing With a Certificate Authority (CA) Signed Certificate

When the **Encrypt** property is set to *true* and the **trustServerCertificate** property is set to *false*, the Microsoft JDBC Driver for SQL Server will validate the SQL Server SSL certificate. Validating the server certificate is a part of the SSL handshake and ensures that the server is the correct server for the connection. In order to validate the server certificate, the trust material must be supplied at connection time either by using the **trustStore** and **trustStorePassword** properties explicitly, or by using the underlying Java Virtual Machine (JVM) default truststore implicitly.

For example:

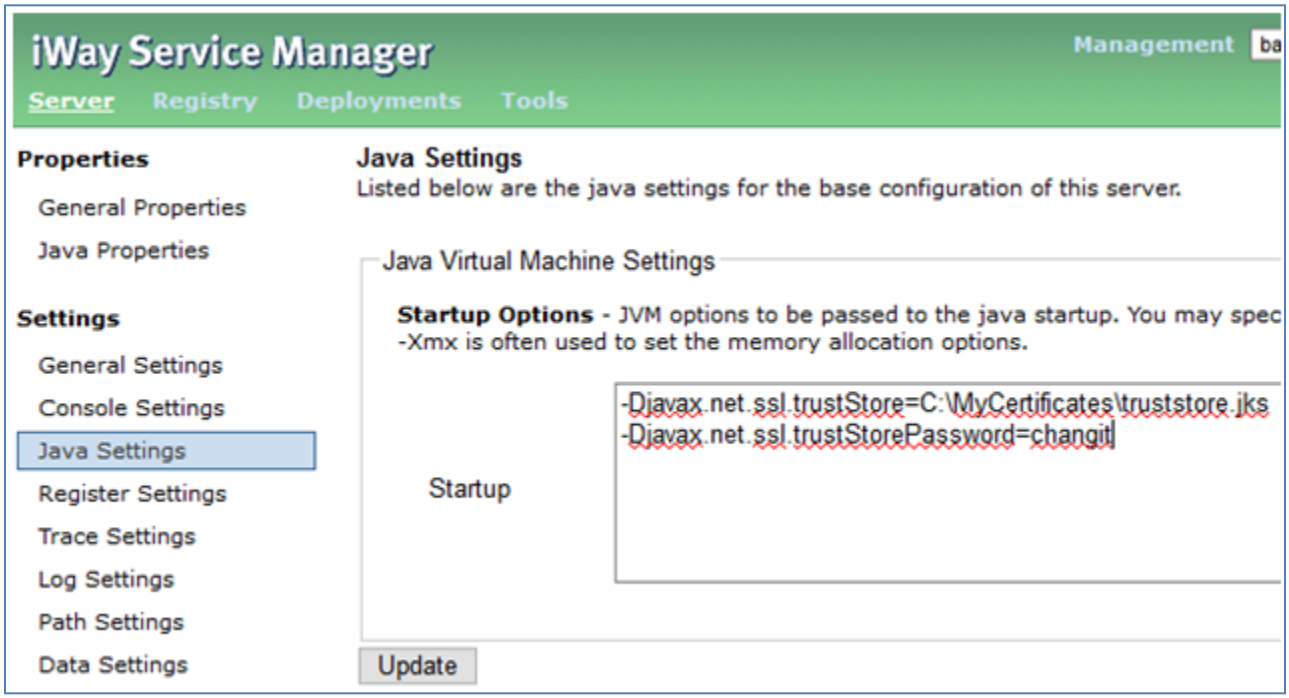
```
-------------------------------------------------------------------------------
jdbc:sqlserver://hostname:1433;databaseName=AdventureWorks2014;encrypt=true;
trustServerCertificate=false;trustStore=C:\Program Files\Java\jdk1.7.0_79\jre\
lib\security\cacerts;trustStorePassword=changeit
-------------------------------------------------------------------------------
```

Alternatively, you can set the following Java Settings (as startup options) within your iWay Service Manager (iSM) Configuration or iWay Integration Application (iIA) Template. These options will apply to all JDBC Data Providers running in the iWay environment.

For example:

```
-------------------------------------------------------------------------------
-Djavax.net.ssl.trustStore=C:\MyCertificates\truststore.jks
-Djavax.net.ssl.trustStorePassword=changit
-------------------------------------------------------------------------------
```

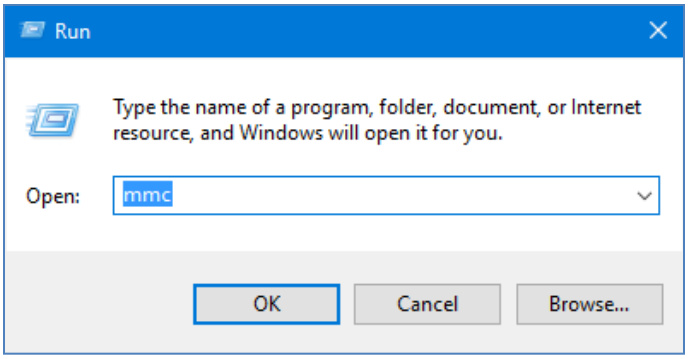
The following is a screenshot of the **Java Settings → Java Virtual Machine Settings (Startup Options)** in the iSM Administration Console:



Exporting the SQL Server Certificate

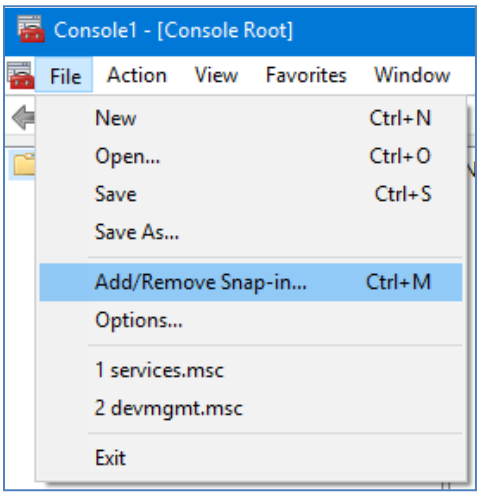
To export the SQL Server certificate:

1. Open the Microsoft Management Console (MMC) by typing *mmc* in the Run dialog and clicking *OK*, as shown in the following image.

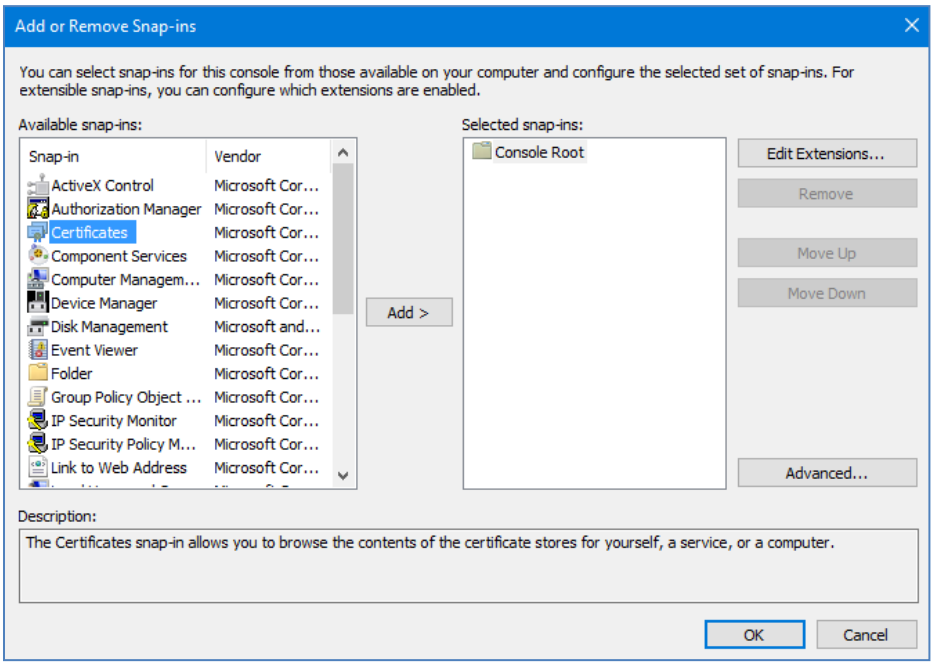


The Microsoft Management Console opens.

2. Click *File* from the menu bar and then select *Add/Remove Snap-in,* as shown in the following image.

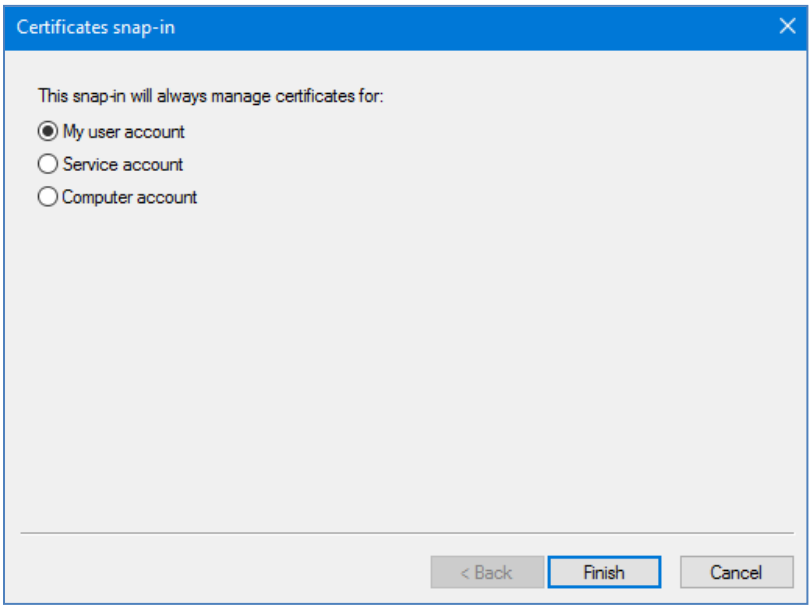


The Add or Remove Snap-ins dialog opens, as shown in the following image.



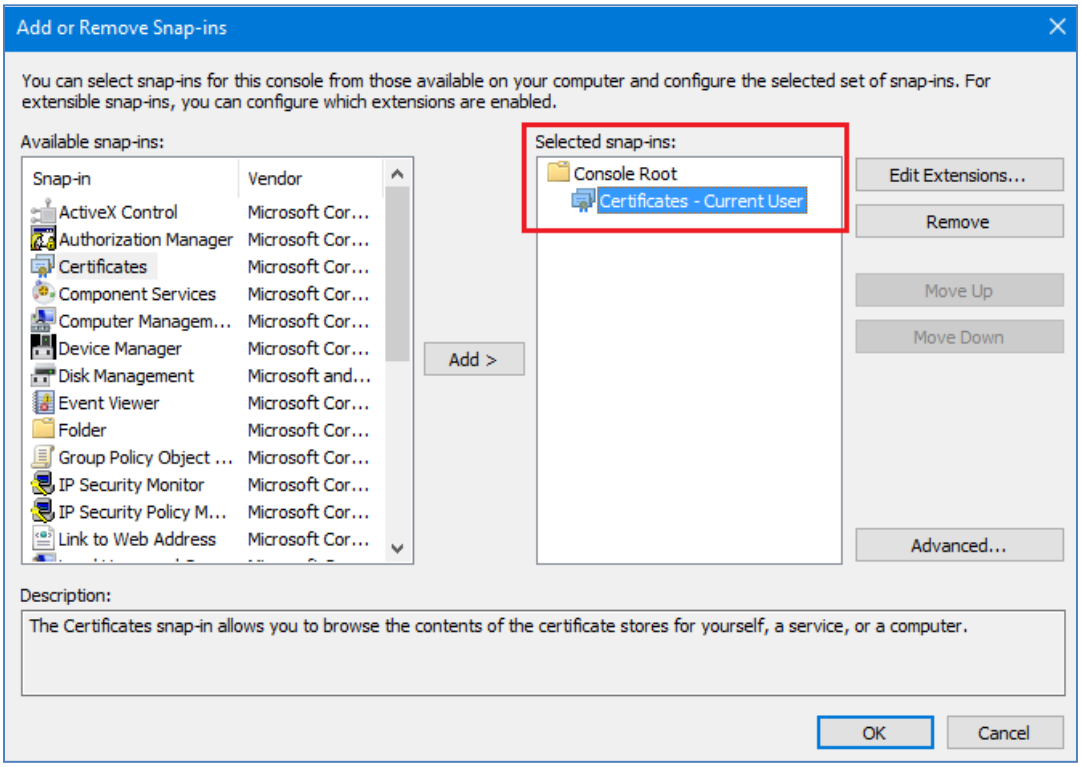
3. In the Available snap-ins area (left pane) of the dialog, click *Certificates*, and then click *Add*.

   The Certificates snap-in dialog opens, as shown in the following image.



4. Ensure *My user account* is selected (default) and click *Finish*.

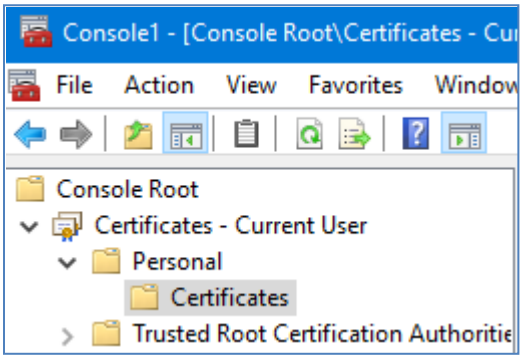
   You are returned to the Add or Remove Snap-ins dialog where the selected snap in (Certificates – Current User) is added to the Selected snap-ins area (right pane), as shown in the following image.



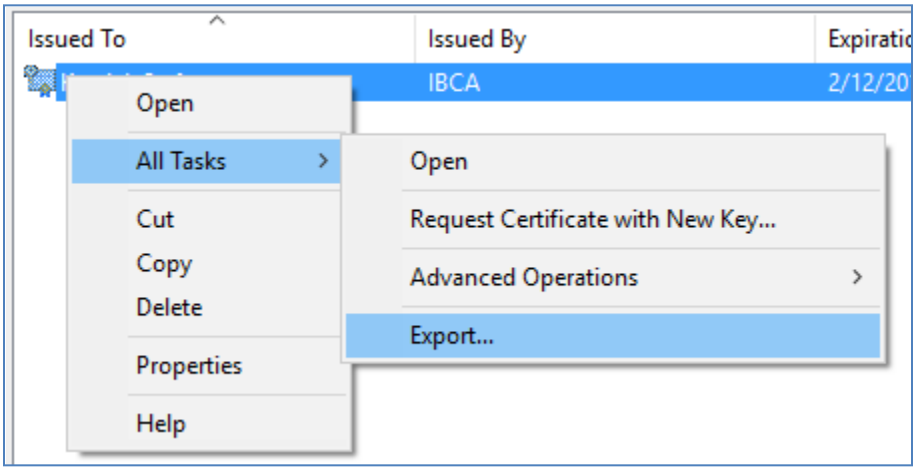
5. Click *OK*.

   You are returned to the main area of the Microsoft Management Console.

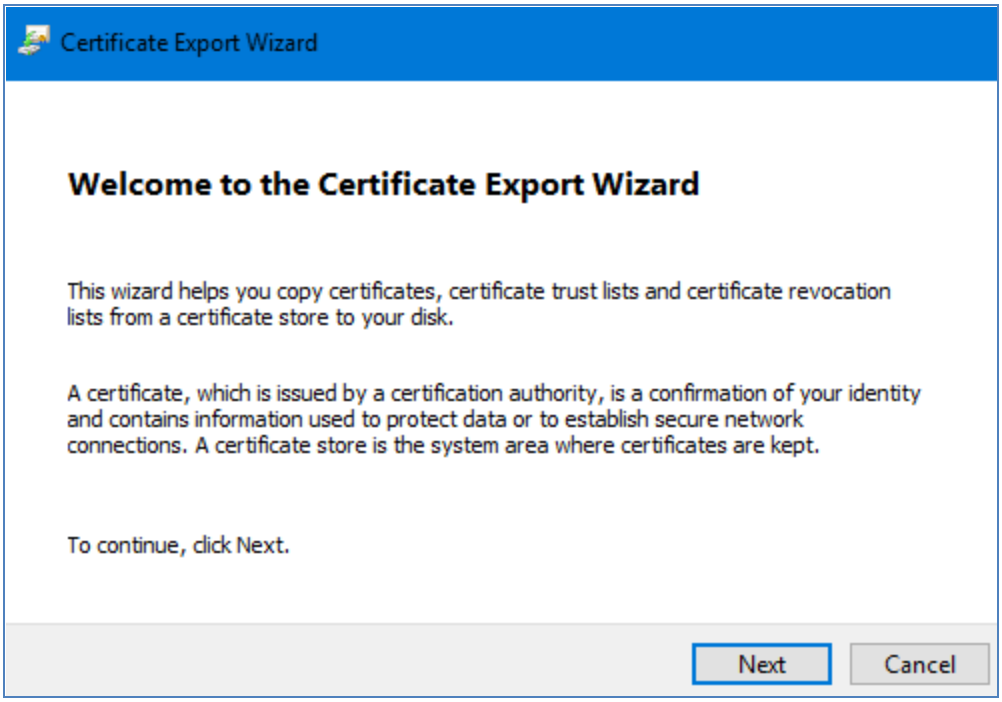
6. In the left pane, expand *Certificates – Current User*, *Personal*, and then click *Certificates*, as shown in the following image.



7. Right-click your certificate in the center pane, select *All Tasks* from the context menu, and then click *Export*, as shown in the following image.



   The Certificate Export Wizard opens, as shown in the following image.

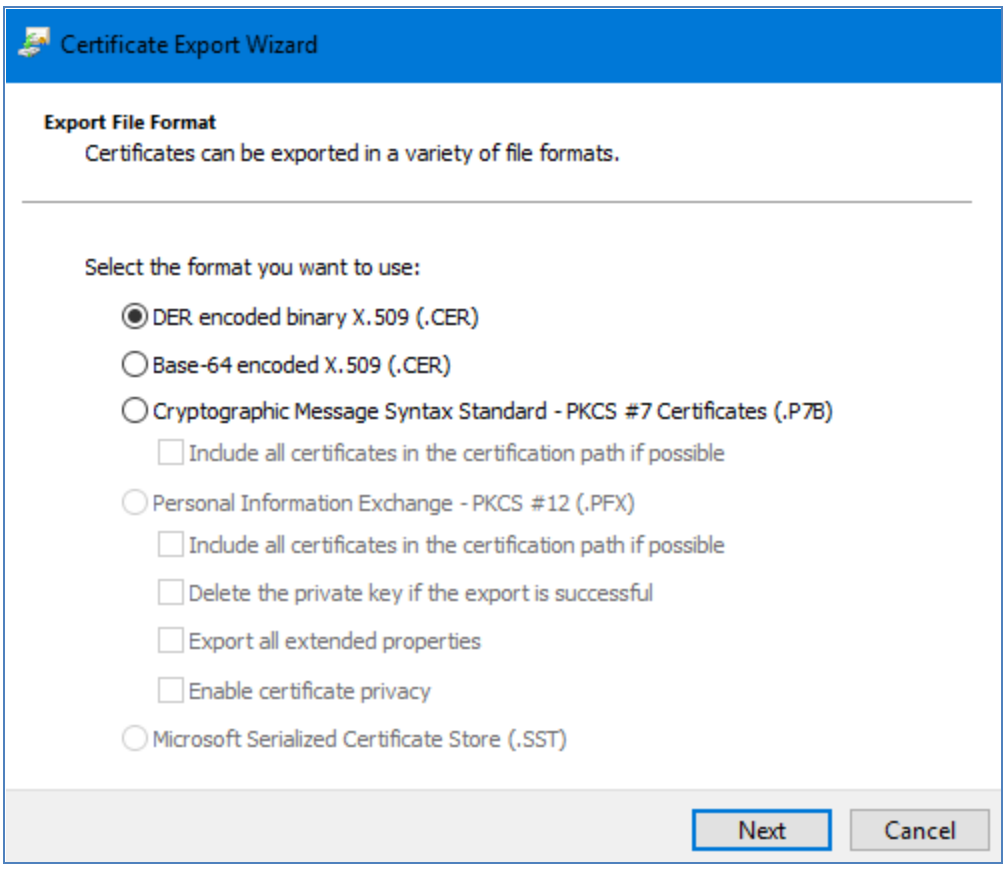


8. Click *Next*.

   The Export Private Key pane opens, as shown in the following image.

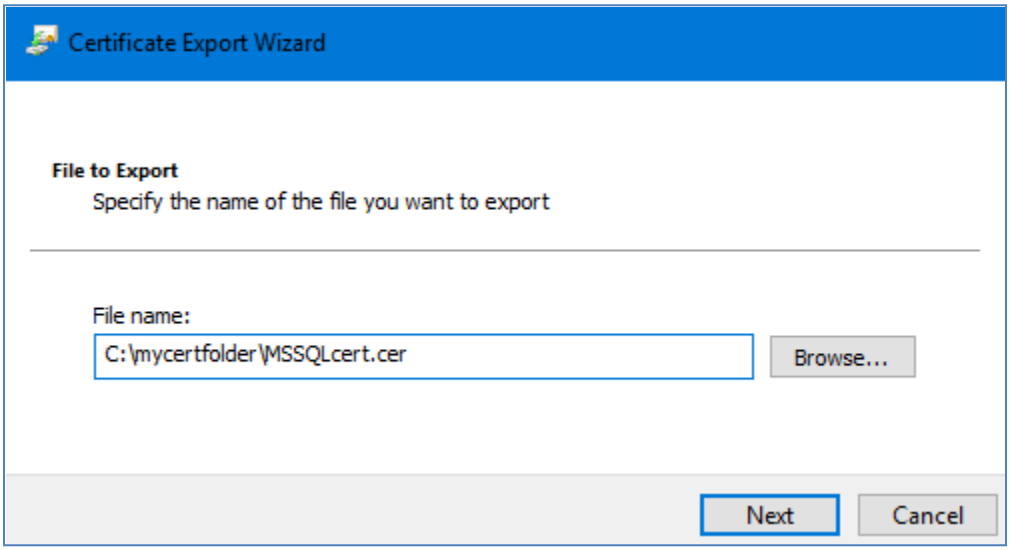
9. Ensure that *No, do not export the private key* is selected (default) and click *Next*.

   The Export File Format pane opens, as shown in the following image.



10. Ensure that *DER encoded binary X.509 (.CER)* is selected (default) and click *Next*.

    The File to Export pane opens, as shown in the following image.



11. Enter (or browse to) the full path for the file you want to create.

12. Click *Next* and then click *Finish.*

This certificate can now be imported into your truststore using the following command:

```
-------------------------------------------------------------------------
keytool -import -v -trustcacerts -alias myServer -file MSSQLcert.cer -
keystore truststore.jks
-------------------------------------------------------------------------
```