Establishing a
secure connection between
ADSelfService Plus and **MS SQL**

Table of Contents

Document summary

ADSelfService Plus supports the external MS SQL database in addition to the bundled PostgreSQL database. This document is intended for admins who want to secure the connection between their MS SQL database and ADSelfService Plus with an SSL certificate. By applying an SSL certificate in the SQL server, you can ensure that the data transferred between ADSelfService Plus and the SQL server is encrypted and stays secure during transmission.

Prerequisites

- You'll need a valid SSL certificate in PFX format that isn't expiring soon. If you have a certificate in another format, please convert it into a PFX file. To create a self-signed certificate using IIS, follow the steps mentioned here.

- The Common Name in the Subject field of the certificate must be the same as the fully qualified domain name (FQDN) of the machine in which the MS SQL Server is installed.

- The certificate must be issued for server authentication, so the Enhanced Key Usage property of the certificate should include *"Server Authentication (1.3.6.1.5.5.7.3.1)."*

Steps to check whether your certificate meets these requirements are listed here.

Important: If you've already applied a valid SSL certificate (matching the requirements in prerequisites) in your SQL Server, you can start on Step 3.
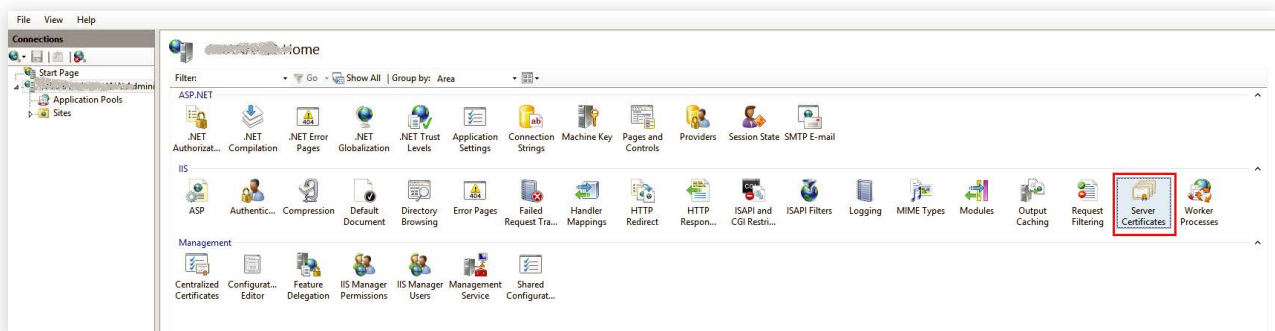
Importing the certificate to the certificate store

If you're using a self-signed SSL certificate generated using Internet Information Services (IIS) Manager, you can start on Step 2.

If you're using a certificate generated through other modes, then you must first import it to the certificate store in SQL Server. You can import the certificate using either IIS Manager or the Microsoft Management Console (MMC) snap-in.

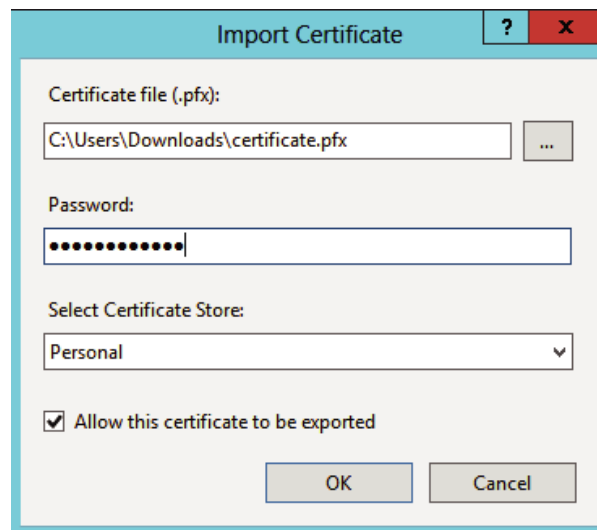## Importing the certificate using IIS Manager

1. Open **IIS Manager.**
2. Click on the name of the server in the **Connections** column in the left pane. In the middle row of icons, double-click on **Server Certificates.**
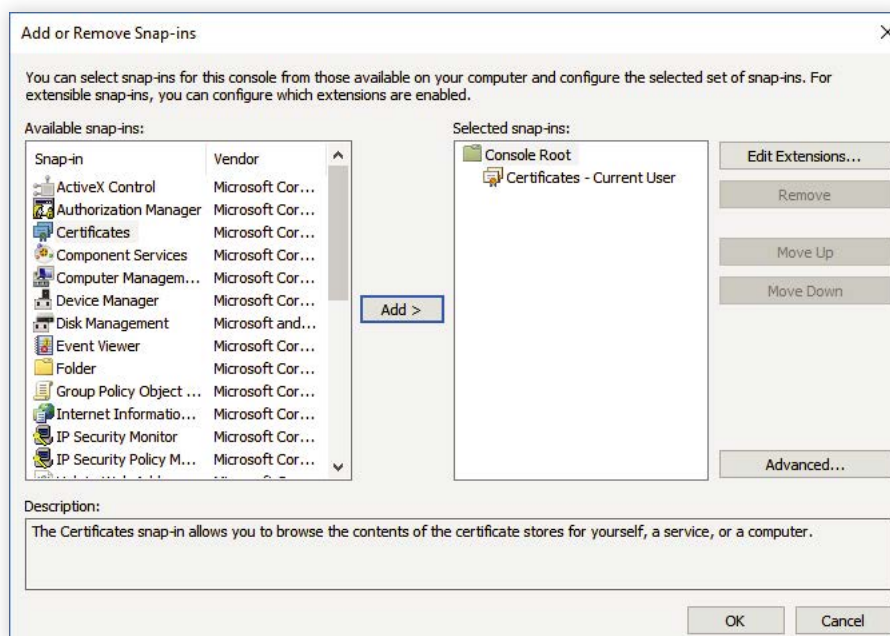

3. Click **Import** in the **Actions** pane.
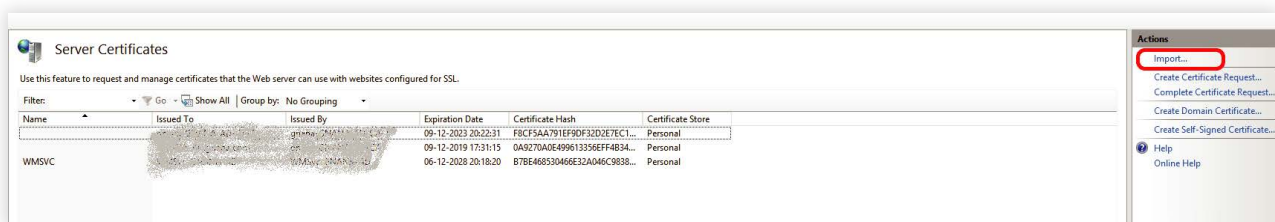
4. Browse and select the **PFX certificate file**.
5. Enter the **password** that you used while generating the certificate file.
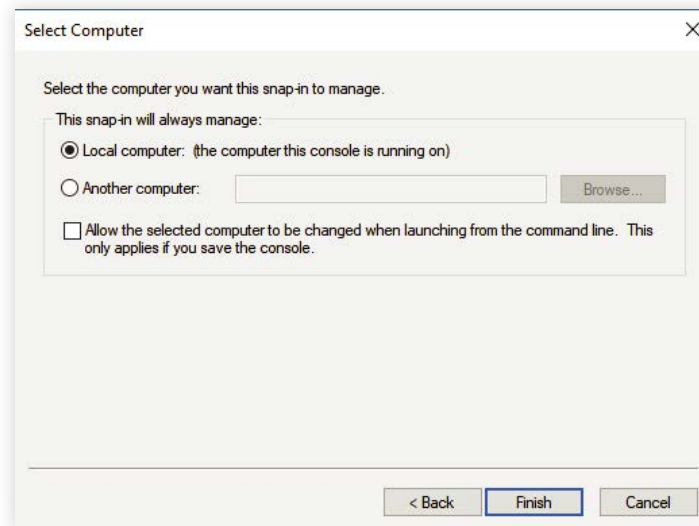
6. Click **OK.**

Importing the certificate using MMC

1. Open **MMC.**
2. From the File menu, click **Add or Remove Snap-in...**
3. Select **Certificates,** and click **Add.**

4. You'll be prompted to open the snap-in for your user account, the service account, or the computer account. Select the **Computer Account.**
5. Select **Local computer**, and then click **Finish.**

6. Click **OK** to exit *Add* or *Remove Snap-in.*

7. Back in the MMC, double-click **Certificates (Local Computer)** to expand the tree view.

8. Right-click on **Personal,** and select **All Tasks > Request New Certificate…**
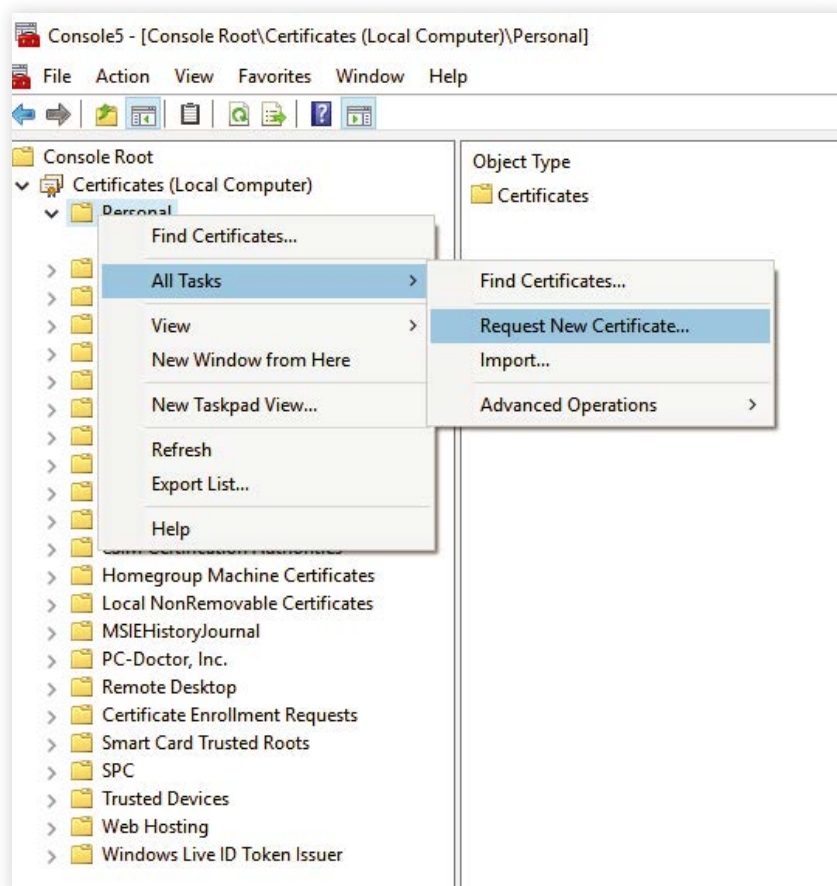
9. Click **Next** in the Certificate Request Wizard that opens.

10. Select **Computer** as the certificate type.
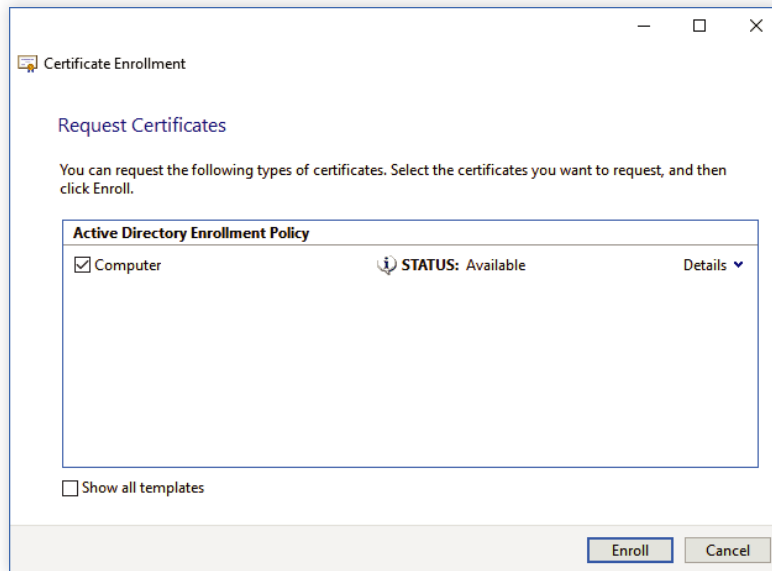
11. You can either enter a name in text box or leave it blank. Then complete the wizard by clicking **Enroll** and then clicking **Finish.**

STEP 2 Associating the certificate with MS SQL Server

1. Open **SQL Server Configuration Manager.**
2. Expand **SQL Server Network Configuration** and right-click on **Protocols** for the MS SQL Server instance to which you want to associate the certificate. Then click **Properties.**
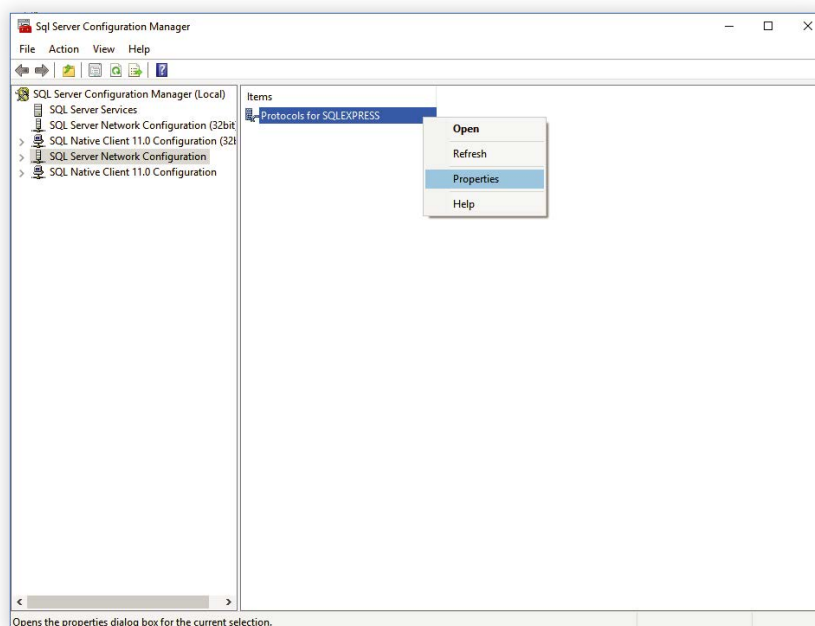
3. On the *Flags* tab, select **Yes** in the *Force Encryption* box.
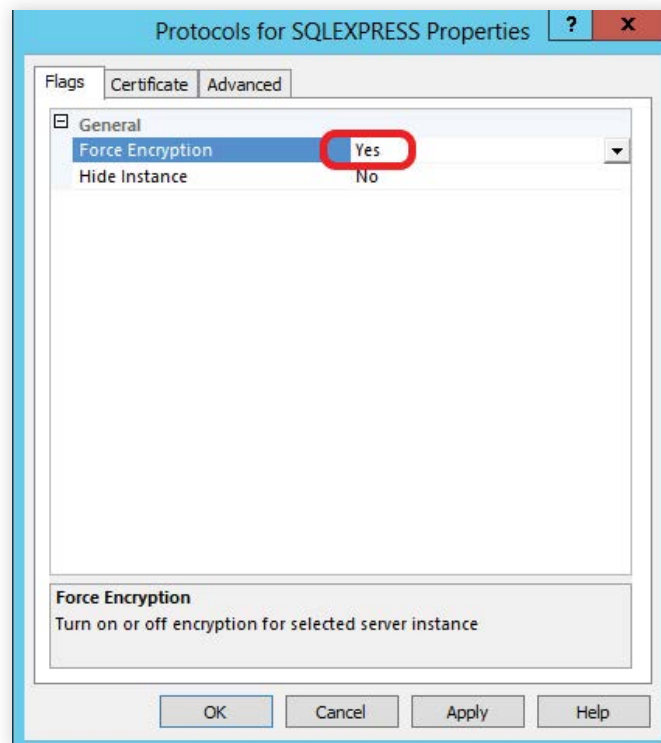
4. On the *Certificate* tab, select the **certificate** you want to use.

5. Click **OK.**
6. Restart **SQL Server.**

Configuring ADSelfService Plus

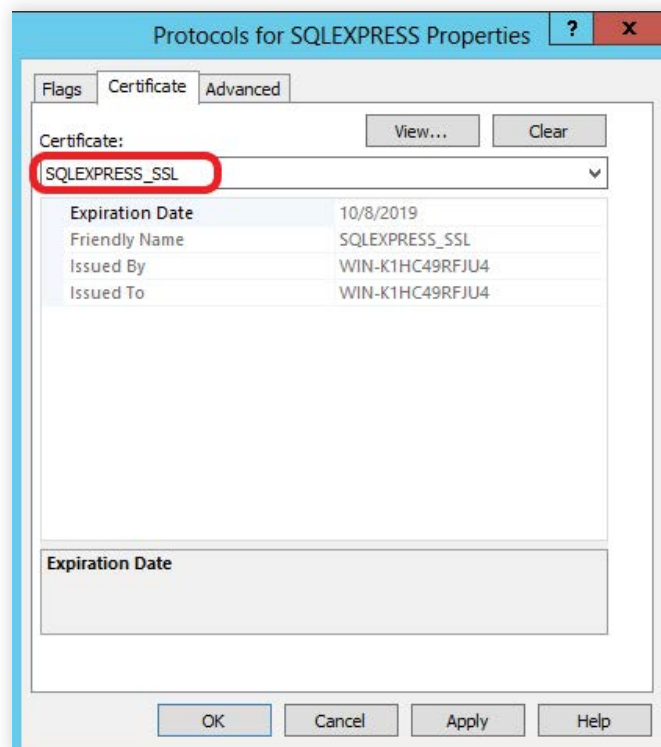After associating the certificate with SQL Server, you need to configure ADSelfService Plus to use the secure connection to the database. Follow the steps below:

1. Go to the **conf** folder in the ADSelfService Plus installation directory (By default: C:\Program Files\ManageEngine\ADSelfService Plus\conf) and open the **database_params.conf** file in a text editor.
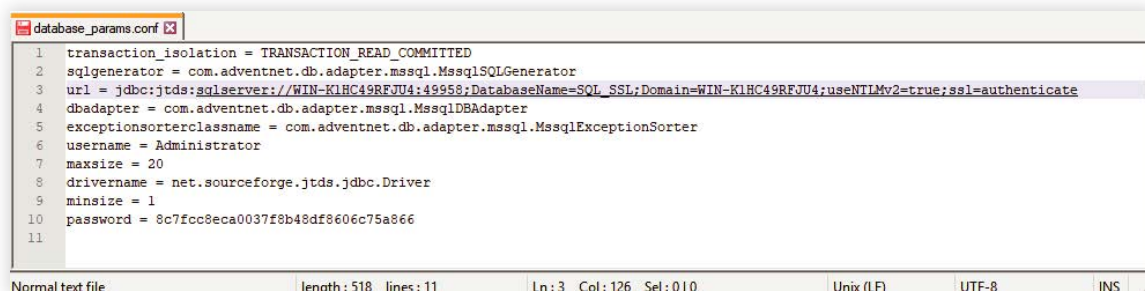

2. You'll see a list of entries such as login, password, and url.
3. Under the **URL** entry, append **ssl=authenticate** to the URL value.

   For example, if the existing entry is:

   url=jdbc:jtds:sqlserver://WIN-K1HC49RFJU4:49958;DatabaseName=SQL_SSL;Domain=WIN-K1HC49RFJU4;useNTLMv2=true

   Then change it to:

   url=jdbc:jtds:sqlserver://WIN-K1HC49RFJU4:49958;DatabaseName=SQL_SSL;Domain=WIN-K1HC49RFJU4;useNTLMv2=true;***ssl=authenticate***

4. In the same **conf** folder, open **wrapper.conf** in a text editor.
5. Search for **wrapper.java.additional.** You'll get a list of entries that are numbered starting at 1.
6. Add the below line after the last wrapper.java.additional entry.

   "wrapper.java.additional.xx=-Djsse.enableCBCProtection=false"

   Where 'xx' denotes the next value to the preceding line's integer.

   For example,

   wrapper.java.additional.1=-Dcatalina.home=..

   wrapper.java.additional.2=-Dserver.home=..

   wrapper.java.additional.3=-Dserver.stats=1000

   ...

   ...

   wrapper.java.additional.12=-DSSL_PORT=8443

   **wrapper.java.additional.13=-Djsse.enableCBCProtection=false**

```
 46   wrapper.java.additional.4=-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
 47   wrapper.java.additional.5=-Djava.util.logging.config.file=../conf/logging.properties
 48   wrapper.java.additional.6=-Duser.home=../logs
 49   wrapper.java.additional.7=-Dlog.dir=../logs
 50   wrapper.java.additional.8=-Ddb.home=../pgsql
 51   wrapper.java.additional.9=-Dcheck.tomcatport="true"
 52   wrapper.java.additional.10=-Dproduct.home=..
 53   wrapper.java.additional.11=-DHTTP_PORT=8888
 54   wrapper.java.additional.12=-DSSL_PORT=9251
 55   wrapper.java.additional.13=-Dfile.encoding=UTF-8
 56   wrapper.java.additional.14=-Djava.io.tmpdir=../temp
 57   #The below line will be executed, when out of memory error occurs.
 58   wrapper.java.additional.15=-XX:OnOutOfMemoryError="SetMaxMemory.bat"    #Call SetMaxMemory.bat file
 59   wrapper.java.additional.16=-Dhaltjvm.on.dbcrash="true"
 60   wrapper.java.additional.17=-Dorg.apache.catalina.SESSION_COOKIE_NAME=JSESSIONIDADSSP
 61   wrapper.java.additional.18=-Dhttps.protocols=TLSv1.2,TLSv1,TLSv1.1
 62   wrapper.java.additional.19=-XX:-CreateMinidumpOnCrash
 63   wrapper.java.additional.20=-Djsse.enableCBCProtection=false
 64   wrapper.java.additional.21=-Djdk.tls.client.protocols=TLSv1.2,TLSv1,TLSv1.1
 65   wrapper.java.additional.22=-Djsse.enableCBCProtection=false
 66
 67   #This filter trigger action will restart the jvm on out of memory error.
 68   wrapper.filter.trigger.1=java.lang.OutOfMemoryError
 69   wrapper.filter.action.1=RESTART
 70   wrapper.filter.message.1=The JVM has run out of memory.
 71
 72   #uncomment the following to enable JPDA debugging
 73   #wrapper.java.additional.3=-Xdebug
 74   #wrapper.java.additional.4=-Xnoagent
```

7. Restart **ADSelfService Plus** to finish.

STEP 4   Associating the certificate to Java Key Store

You need to associate the certificate with the ADSelfService Plus Java KeyStore to establish trust.
Follow the steps given below in the machine in which ADSelfService Plus is installed:

1. Open **IIS Manager.**
2. In the middle pane, click **Server Certificates.**
3. Open the certificate that you want to use.
4. Click on the **Details** tab.
5. Click **Copy** to file.
6. Click **Next** in the Certificate Export Wizard that opens.
7. In the *Export Private Key* screen, select No, **do not export the private key,** and click **Next.**
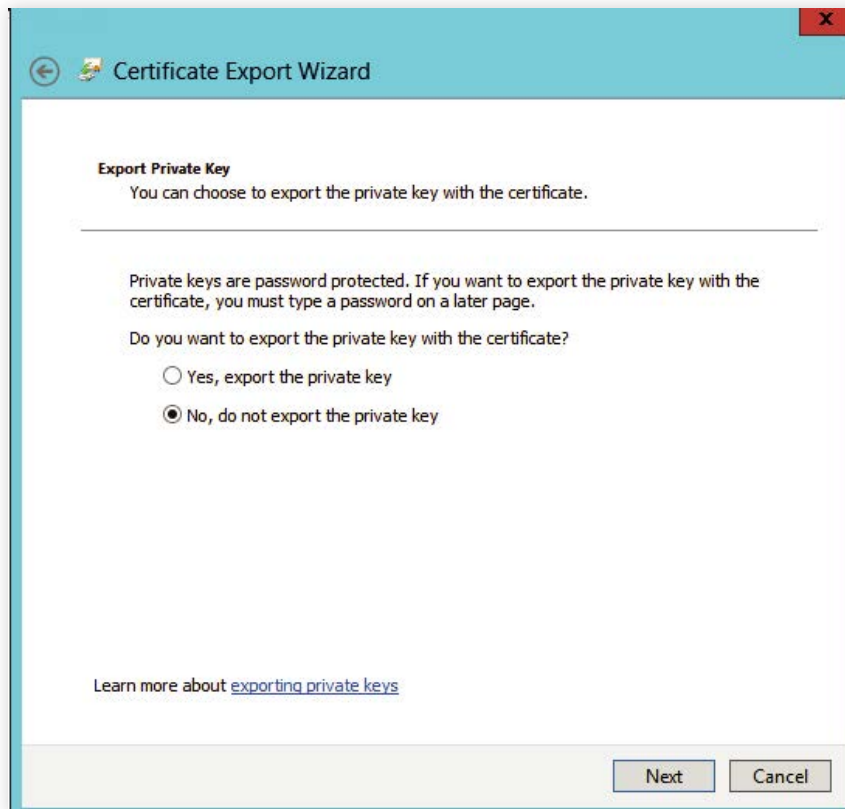
8. In the *Export File Format* screen, select either **DER encoded binary X.509 (.CER)** or **Base-64 encoded X.509 (.CER)**, and click **Next.**
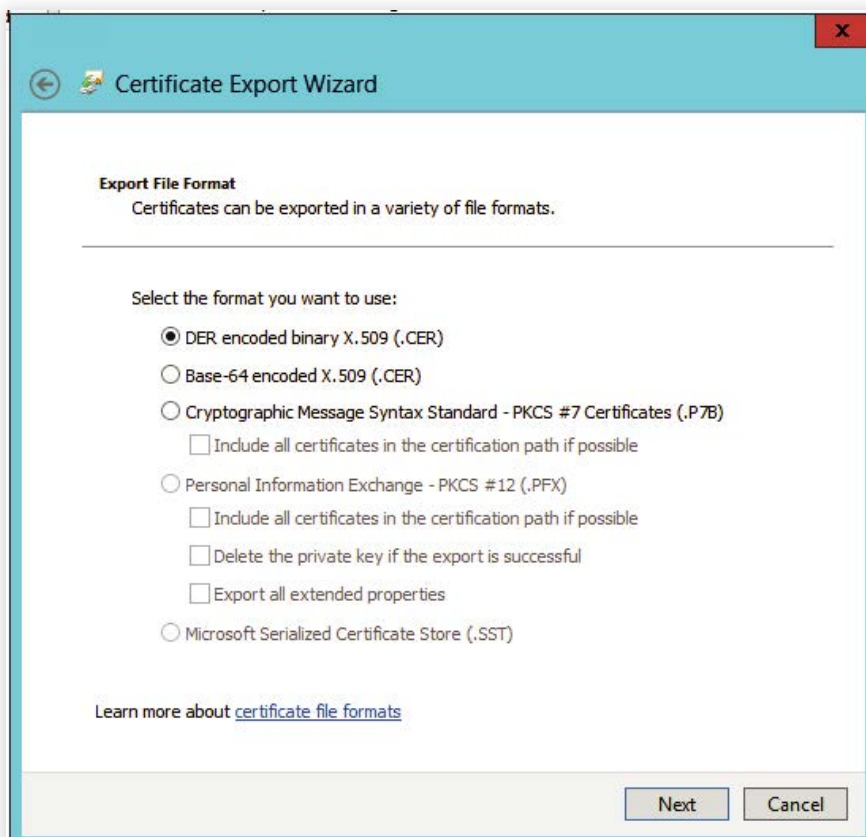
9. Enter a name for the file, and click **Next.**

10. Click **Finish.**

11. Go the the **jre** folder in the ADSelfService Plus installation directory (By default: C:\Program Files\ManageEngine\ADSelfService Plus\jre). Open the **bin** folder.

12. Open Command Prompt and run the following command:

```
"%ADSelfService_Plus_Installation_directory\jre\bin\keytool" -import-v-trustcacerts
-alias myserver -file ssl.cer -keystore

"%ADSelfService_Plus_Installation_directory\jre\lib\security\cacerts" -keypass changeit
-storepass changeit -noprompt
```

Where **"%ADSelfService_Plus_Installation_directory\jre\bin\keytool"** specifies the location of the **keytool.exe** utility, which is used to manage keys and certificates.

Replace this path with the ADSelfService Plus installation directory, which by default is: *C:\Program Files\ManageEngine\ADSelfService Plus\jre\bin.*

Appendix

SSL encryption for failover clustering in SQL Server

If you'd like to use encrypted connections in a clustered environment, then you should have a certificate issued to the fully qualified DNS name of the failover clustered instance. Also, this certificate should be installed on all of the nodes in the failover cluster. Additionally, you'll have to edit the thumbprint of the certificate in the registry because it's set to Null in a clustered environment

The following steps should be performed on all of the nodes in the cluster:

1. Open the **certificate** using the *MMC Certificates Snap-in.*
2. Copy the **hex value** from the *Thumbprint property* on the Details tab to Notepad and remove the spaces.
3. Start **regedit** and copy the **hex value** to this key: HKLM\SOFTWARE\Microsoft\Microsoft SQL Server\<YourSQLServerInstance>\MSSQLServer\SuperSocketNetLib\Certificate.
4. You'll now have to reboot your node, so it's recommended that you failover to another node first.
5. Repeat this procedure on all nodes.

Creating self-signed certificates using IIS

1. Open **IIS Manager.**
2. Click on the **server name** in the *Connections* column in the left pane.
3. Double-click on **Server Certificates** in the middle pane.
4. Click on **Create Self-Signed Certificate** in the Actions column on the right.
5. Enter a name, and click **OK** to proceed.
6. Click **OK.**

You should now see the **SSL** certificate valid for one year.

Checking for SSL certificate validity

1. Open **MMC.**

2. On the *File* menu, click **Add or Remove Snap-in...**

3. Select **Certificates** and click **Add.**

4. You'll be prompted to open the snap-in for your user account, the service account, or the computer account. Select the **Computer Account.**

5. Select **Local computer,** and then click **Finish.**

6. Click **OK** to exit *Add or Remove Snap-in.*

7. Back in the MMC, open the **Certificates** snap-in.

8. Double-click on **Personal** and then **Certificates.**

9. In the right pane, locate the **certificate** you're going to use.

10. The value for the *Intended Purpose* column must be **Server Authentication.**

11. The value for the *Issued* To column must be the **server name.**

12. Now, double-click the **certificate** to view its properties.

13. Under the *General* tab, you should be able to view the message *"You have a private key that corresponds to this certificate."*

14. Under the *Details* tab, the value of the *Subject* field must be the **server name.**

15. The value for the *Enhanced Key Usage* field must be **Server Authentication (1.3.6.1.5.5.7.3.1).**

16. Under the *Certificate Path* tab, the **server name** must appear under the *certification path.*