

TLS 1.2 Support on UC Suite's MS-SQL Server

Applies to:

Imagicle UC Suite rel. 2020.Winter.1 and above

Description:

This article explains the necessary steps to enable TLS 1.2 secure connection to SQL Server. This is highly recommended if the SQL Server instance is running on a different server.

Check Certificate requirements

If you want to use a secure connection to SQL Server, a valid certificate must be used by SQL Server.

If you already have a trusted certificate for the Imagicle Server please skip this session. Otherwise you can build a self-signed certificate suitable for a SQL Server in a lab/test environment, by following this procedure:

- Login onto Imagicle Application Suite
- Open a command prompt and move into the following Directory:
<StonevoiceAS>\System\SSL
- Launch the following command:

```
makecert -r -pe -n "CN=MININT-Q99PLQN.fareast.corp.microsoft.com" -b 10/16/2015 -e 12/01/2020 -eku 1.3.6.1.5.5.7.3.1 -ss my -sr localMachine -sky exchange -sp "Microsoft RSA SChannel Cryptographic Provider" -sy 12 -a sha256
```

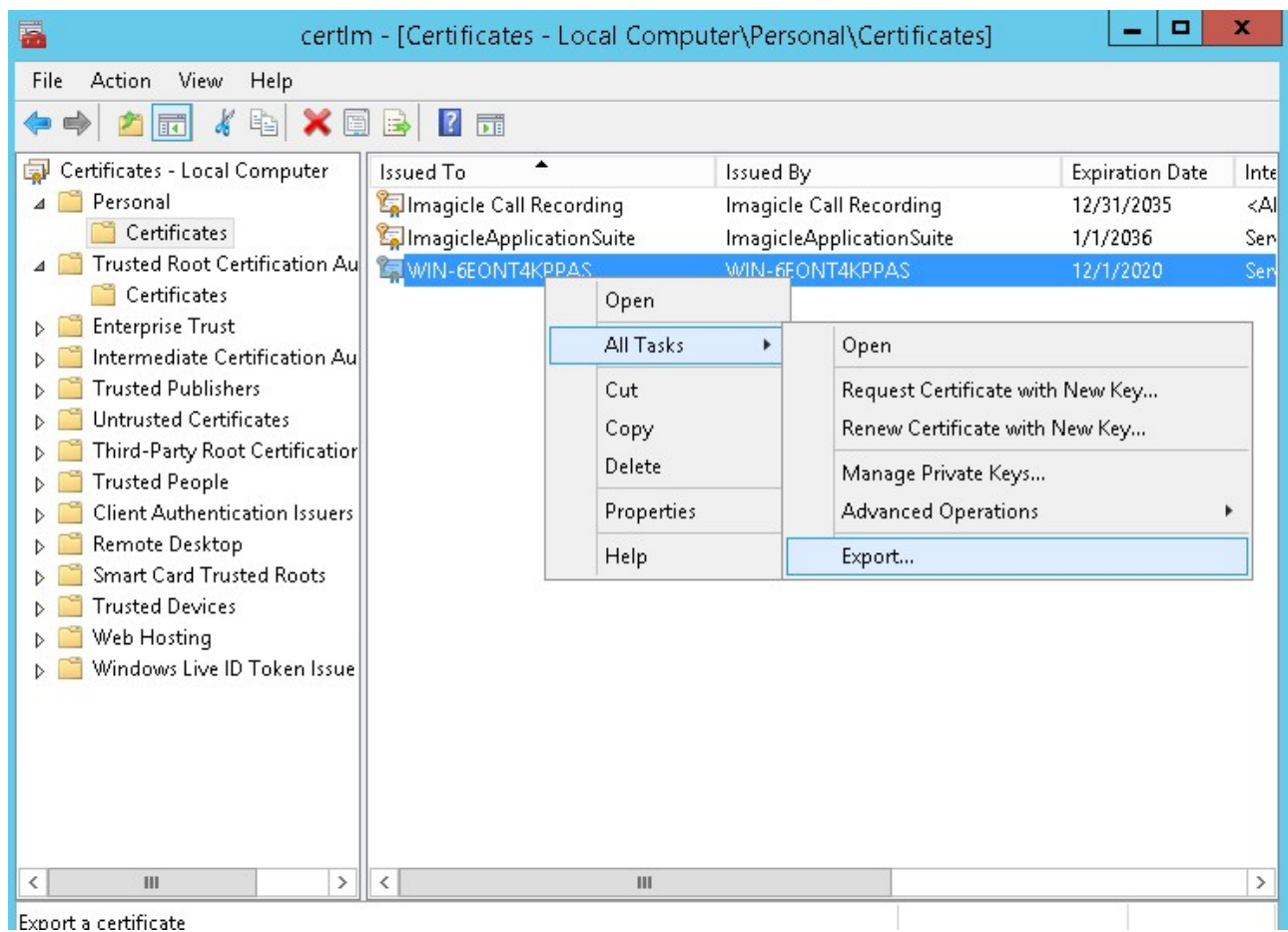
- The name of the certificate (CN=) must equal the server FQDN or full computer name of Imagicle Server

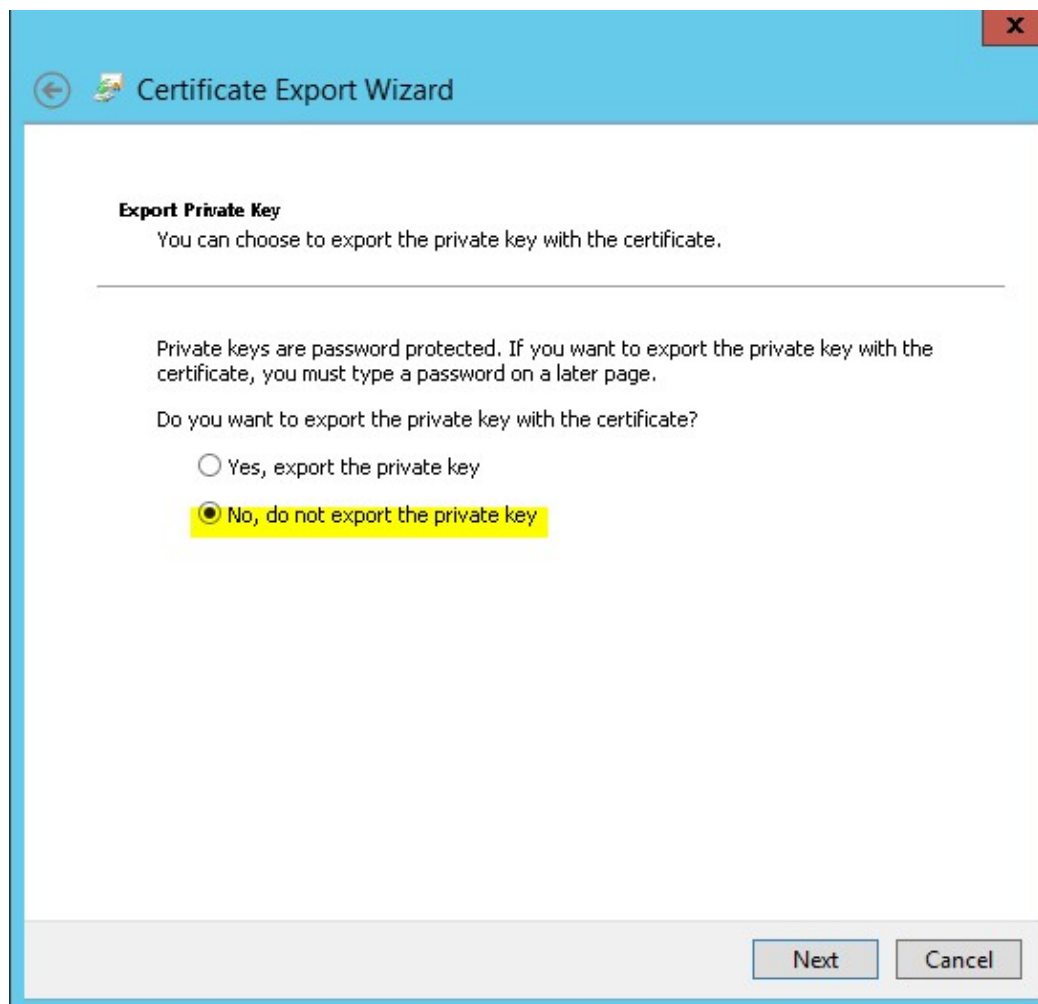
Following mandatory requirements of SQL certificate:

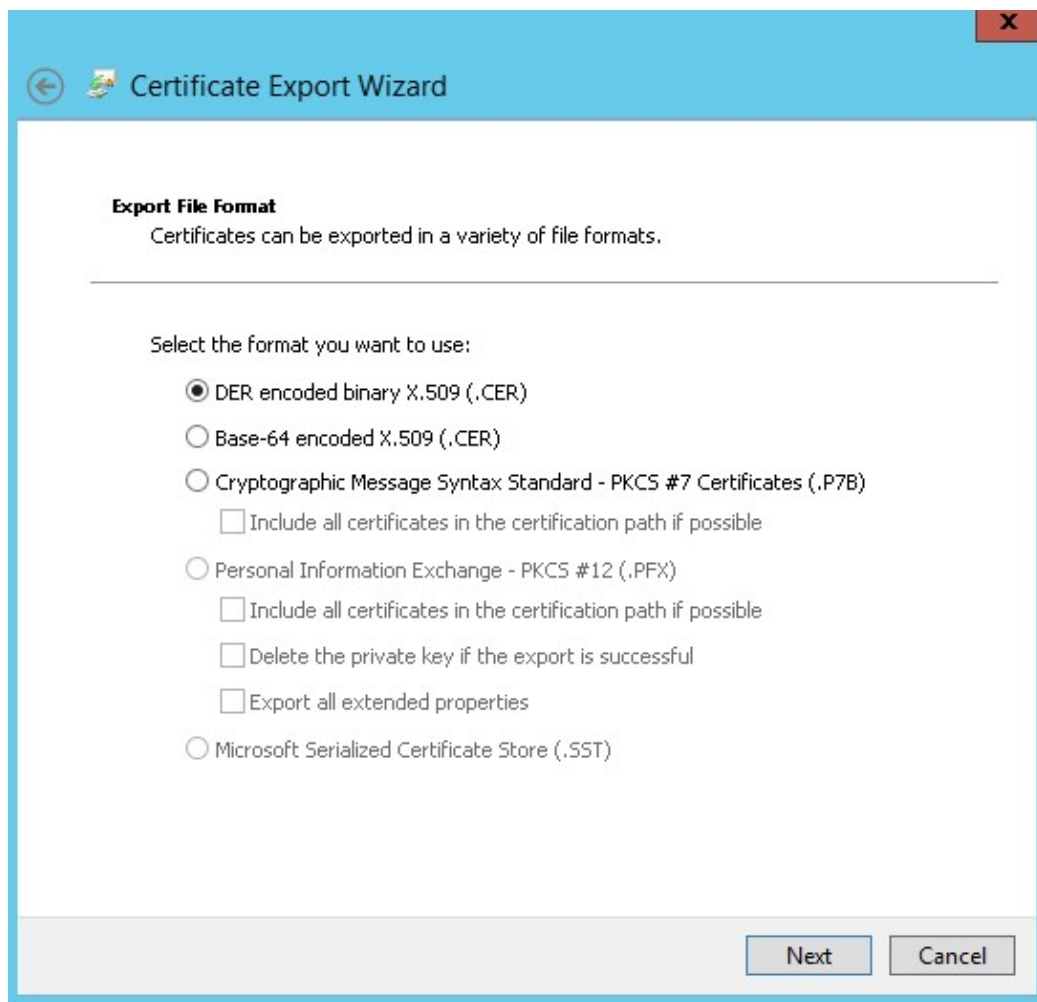
- It must be valid thus the current system date and time should be between the Valid From and Valid To properties of the certificate.
- It must be available into WCS "Personal" section (Computer account)
- The Common Name (CN) in the Subject property of the certificate must be the same as the fully qualified domain name (FQDN) of the server computer.
- It 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)**'.
- It must be created by using the KeySpec option of 'AT_KEYEXCHANGE'.

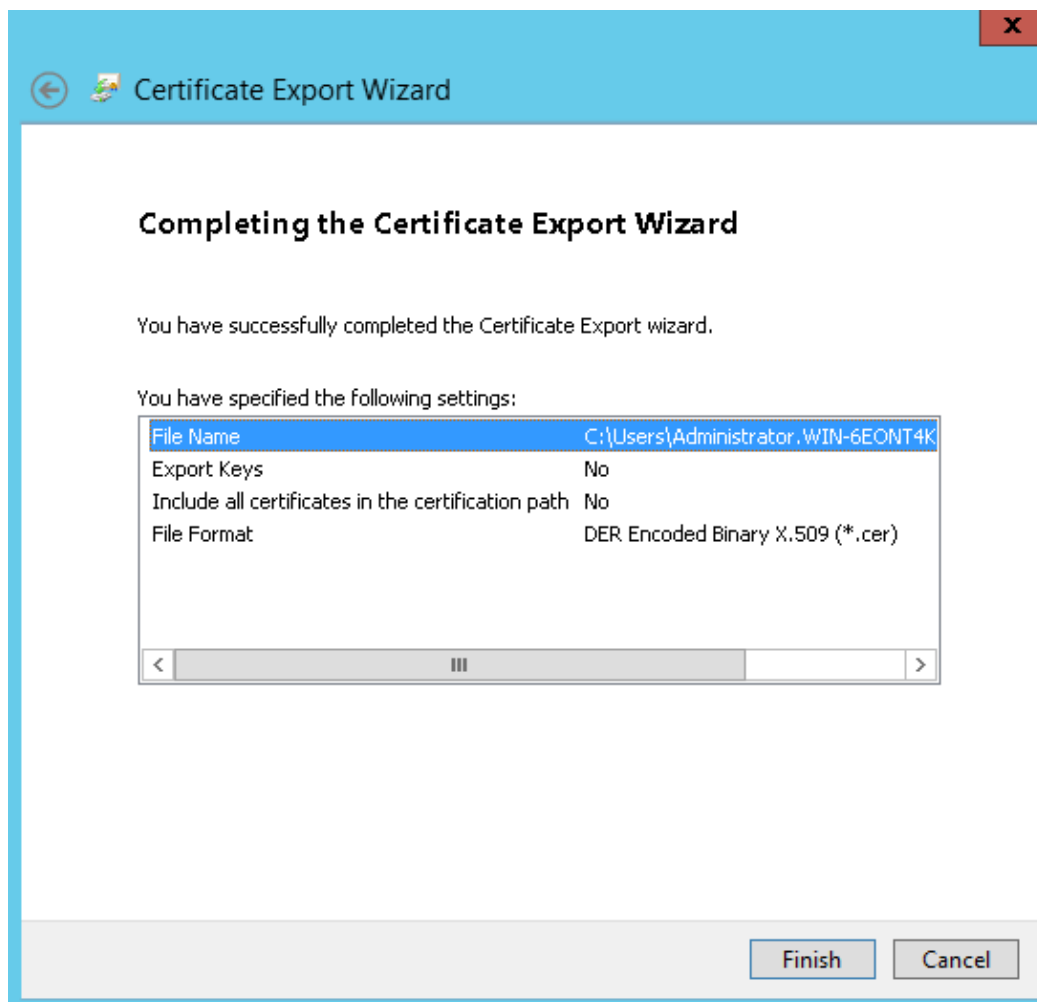
Moreover, the certificate should be available in "Trusted root certification authorities". If not available, you can export it without a private key from **Personal** à **Certificates** and subsequently import it in **Trusted Root Certification Authorities** à **Certificates**. See below screenshots for the complete export/import procedure:

Certificate Export



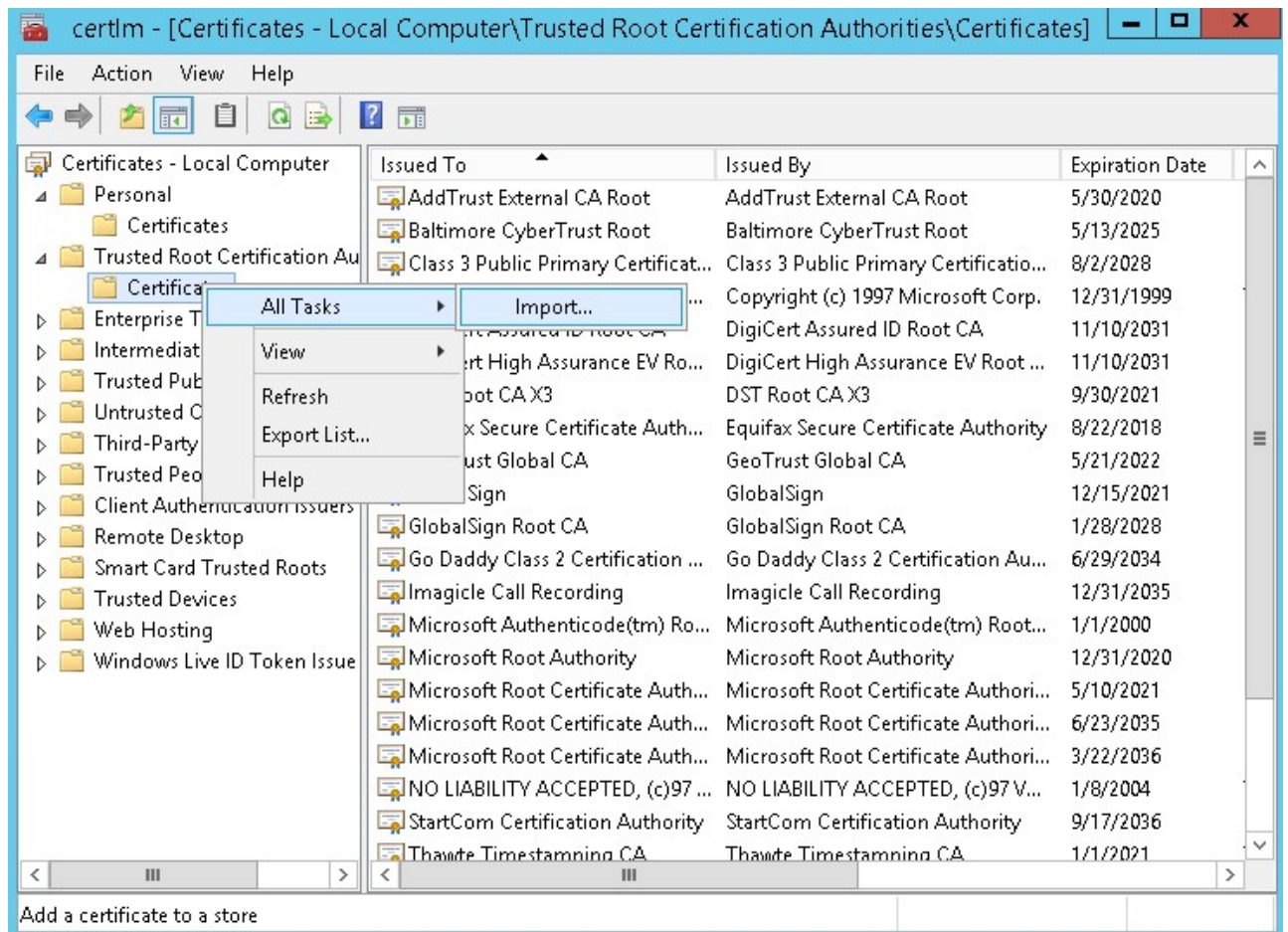









Certificate Import

Import must be done on the machine where SQL Server instance is installed.









Certificate Import Wizard

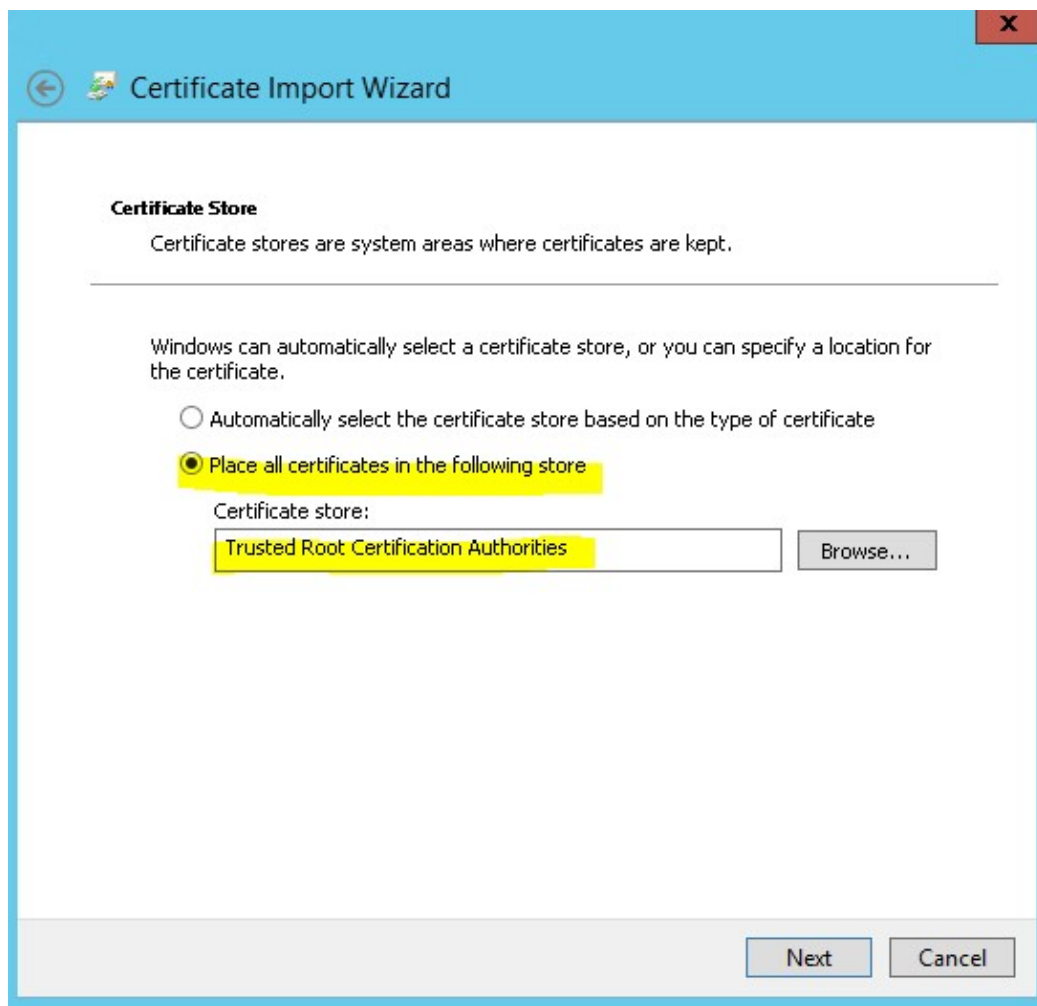
File to Import

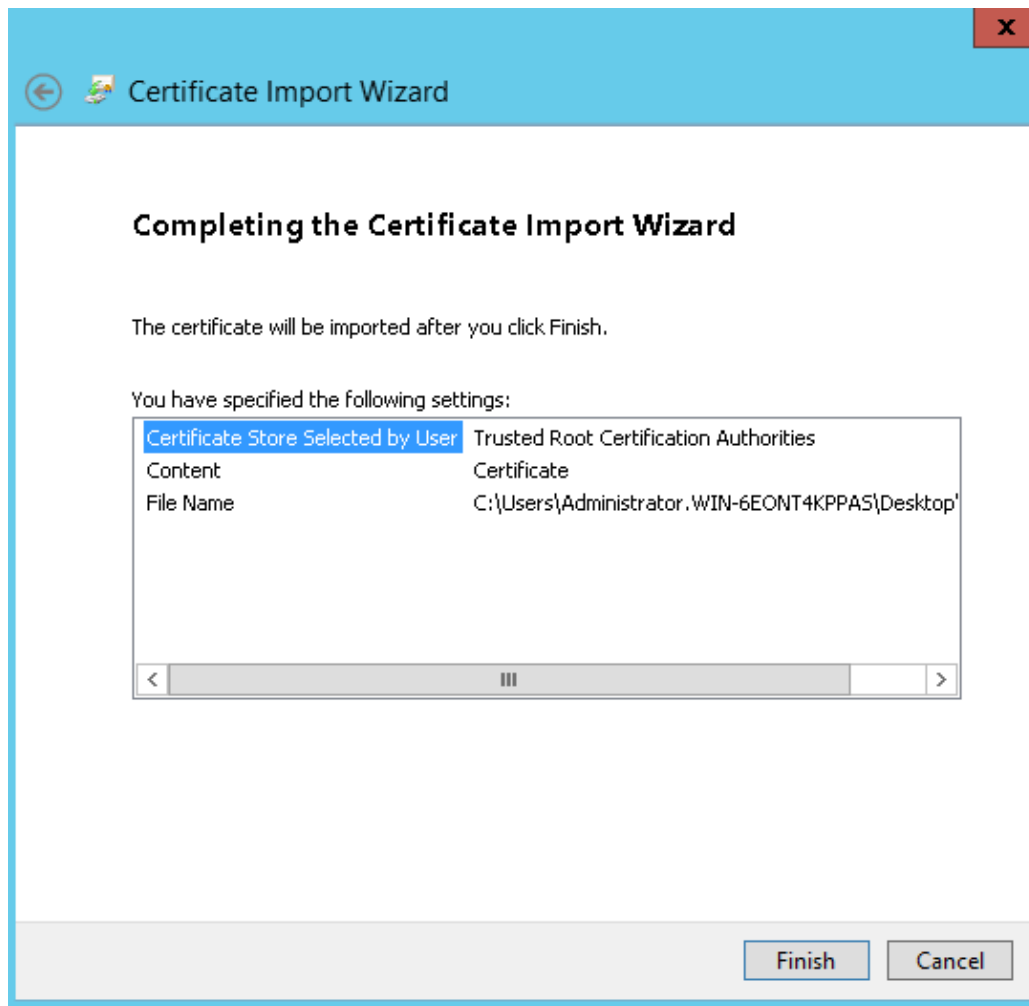
Specify the file you want to import.

File name:

Note: More than one certificate can be stored in a single file in the following formats:

- Personal Information Exchange- PKCS #12 (.PFX, .P12)
- Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)
- Microsoft Serialized Certificate Store (.SST)



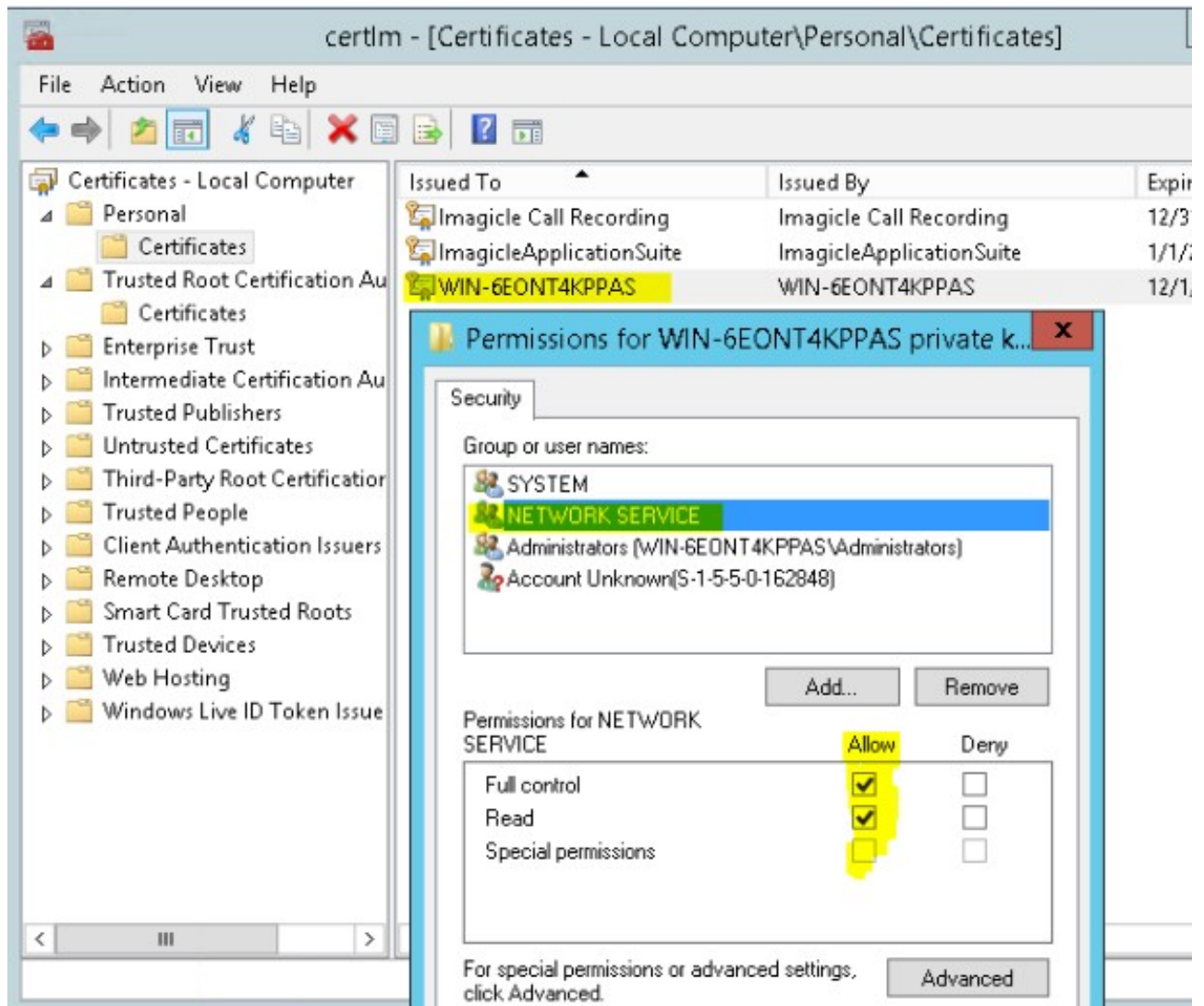


See [this link](#) for additional information.

Please notice that these requirements do not apply if you're going to establish a plain (unsecure) connection to SQL Server.

Also ensure the following:

- The certificate must be listed in "Personal" section of WCS (check using certlm.msc)
- The "Subject" property must equal server FQDN
- Server authentication (eku=1.3.6.1.5.5.7.3.1) must be enabled
- Must be created with KeySpec option set to "AT_KEYEXCHANGE"
- Must also be listed in "Trusted root certification authority" section: if not listed, copy from "Personal/Certificates" section .
- Check certificate permissions: "NETWORK SERVICE" user must be present and have Full control

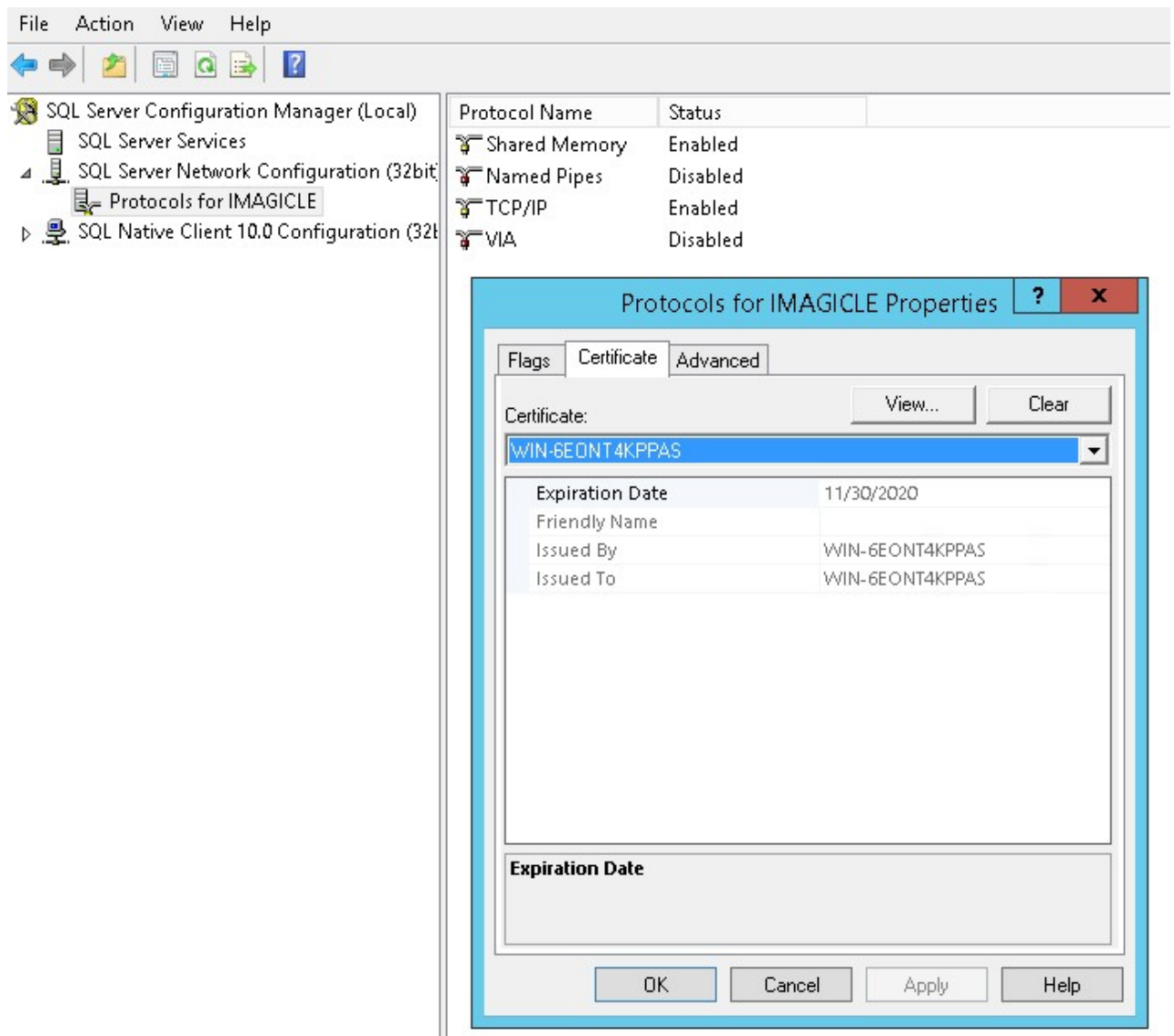


SQL Server Engine Configuration

To allow encrypted connections to SQL Server, you must configure a certificate. This is accomplished in two different ways:

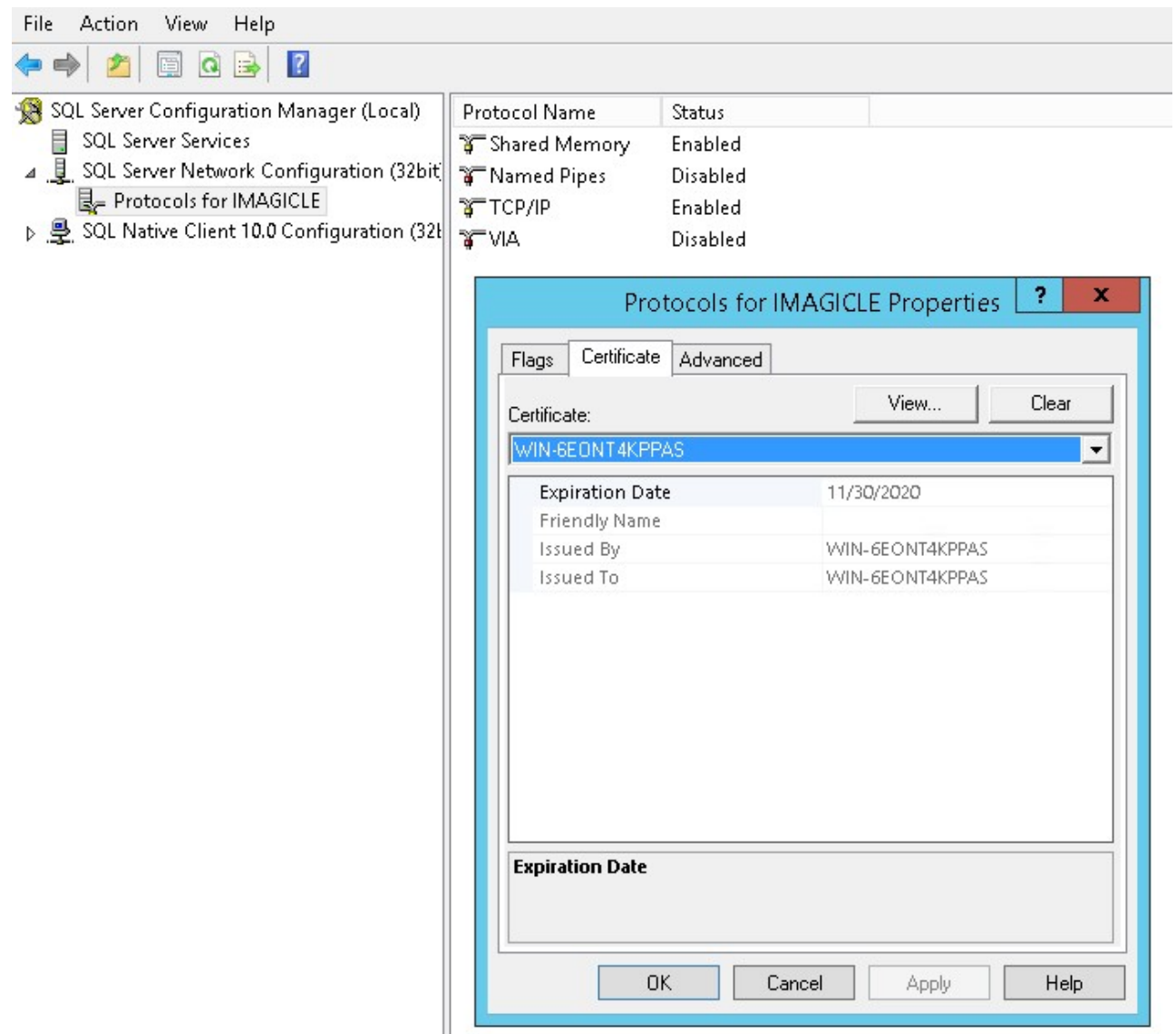
1. Using own trusted certificate (in production environments):

- Pls. start SQL Server Configuration Manager and select **"SQL Server Network Configuration"** → **"Protocols for IMAGICLE"**. Right-click on **"Properties"** and select **"Certificate"** tab. Here you can add your own trusted certificate.



2. Using a self-signed, auto-generated user's certificate (test environments):

- Pls. start SQL Server Configuration Manager and select "SQL Server Network Configuration" > "Protocols for IMAGICLE". Right-click on "Properties" and select "Certificate" tab. Here you can add new self-signed certificate (check certificate requirements for SQL Server).



Configure the UC Suite to use secure connection to SQL Server

Secure connection to SQL server is not mandatory for TLS setup. However, it is recommended when SQL server runs on a different server.

If you want to use a secure connection to the SQL Server, run the Imagicle Database Configuration tool (from Start Menu/Imagicle UC Suite), then select the "Use secure connection" checkbox and complete the procedure following the configuration wizard's instructions.

If an external SQL Server is used, the FQDN must be entered in the SQL Server location

Adjust the UC Suite SQL client version

Regardless you are using or not a secure connection to SQL, you need to increase the SQL client version used by the Imagicle services to connect to the database:

1. Edit the file StonevoiceAS\System\SvSasDB.ini and replace the word 'SQLNCLI10' with 'SQLNCLI11'.
2. Save the file.
3. Stop and Start all Imagicle services or restart the server.

Complete HA-related configurations

In case of an HA environment, ensure all servers have all cluster certificates imported.