

Configuring HTTPS and custom ports

You can configure SQL Clone Server to serve the web UI over HTTPS and/or a custom port by creating a settings file.

Configure SQL Clone Server

1) If it doesn't already exist, create a new text file named **SharedAppSettings.config** in SQL Clone Server's installation directory and paste the following into it:

```
<?xml version="1.0" encoding="utf-8"?>
<appSettings>
  <add key="ClientEndpointUseHttps" value="false" />
  <add key="ClientEndpointPort" value="14145" />
</appSettings>
```

2) If you want to use a custom port, set the value of **ClientEndpointPort** to the required port number. (14146 is in use to communicate with the Clone agents, and shouldn't be used.)

3) If you want to use HTTPS, set the value of **ClientEndpointUseHttps** to "true" and refer to the next guide on this page to configure HTTPS for your chosen port.

4) Save **SharedAppSettings.config** and restart the SQL Clone Server service using **services.msc** or your preferred tool.

Configure Windows to support HTTPS

You only need to do this section if you want to use HTTPS.

Obtain a certificate

You can either use an SSL certificate issued by a legitimate Certificate Authority (CA), or create a self-signed SSL certificate with a self-created CA.

If you use an SSL certificate from a legitimate CA, you will only need the certificate's .pfx (private key) file.

If you use a self-signed SSL certificate with a self-created CA, you will need the SSL certificate's .pfx file as above, but also the self-created CA's .cer file.

Add the certificate to the certificate store

1) Open **MMC.exe** with administrative privileges.

2) In **File, Add/Remove Snap-in**, add the Certificates snap-in. During that process, select **Computer Account**, and then **Local Computer**.

3) Right-click on **Personal, All Tasks, Import**. Locate the SSL certificate's .pfx file. If it has a password, enter it. Proceed through the process until it reports that the import was successful.

4) Find the imported certificate in **Personal/Certificates**. Double-click on it, and open the **Details tab**. Locate the **Thumbprint field** - you will need the thumbprint value for the next section.

If you used an SSL certificate from a legitimate CA, then proceed to the next section. If you used a self-signed certificate from a self-created CA, then the SSL certificate will not yet be trusted by your computer, because the certificate has been signed by a CA it does not know or trust. Proceed on to step five to make your computer trust the self-created CA.

5) Right-click on **Trusted Root Certification Authorities, All Tasks, Import**. Locate the self-created CA's .cer file. Proceed through the process until it reports that the import was successful. *This step will have to be repeated on all computers that will use SQL Clone Server's web UI.*

Modify the network configuration to use the certificate for SQL Clone Server

1) Open **cmd.exe** with administrative privileges. Open **netsh** with netsh, then its **http** mode with http.

2) Run this **urlacl** command to allow SQL Clone Server's user account to serve requests on your desired URL. Make sure to change the port number in url to your desired port number, and SOMEDOMAIN\SqlCloneServerAccount to the user account configured for SQL Clone Server's service.

```
add urlacl url=https://+:14145/ user=SOMEDOMAIN\SqlCloneServerAccount
```

3) Run this **sslcert** command to have Windows use the certificate you created to secure connections on your desired URL. Make sure to change the port number in **ipport** to your desired port number, and **certhash** to the SSL certificate's thumbprint (not the CA's) with no spaces or stray characters. You can also change the **appid** to a new GUID (it doesn't need to correspond to anything else).

```
add sslcert appid={eca24963-e976-4fd4-a071-a819a7c7a4ac} ipport=0.0.0.0:14145
certhash=18f7475a7c729531b29c25414c682e6e11737aed
```