Technical Note

Secure AlwaysOn Connections between Kepware and ThingWorx

Safety and uptime are key components of industrial automation and industrial IoT production systems, such as discrete manufacturing, or oil and gas production. Kepware® products enable communications for these systems via the AlwaysOn protocol. With cybersecurity threats increasing in both frequency and complexity, it is paramount to secure these connections.

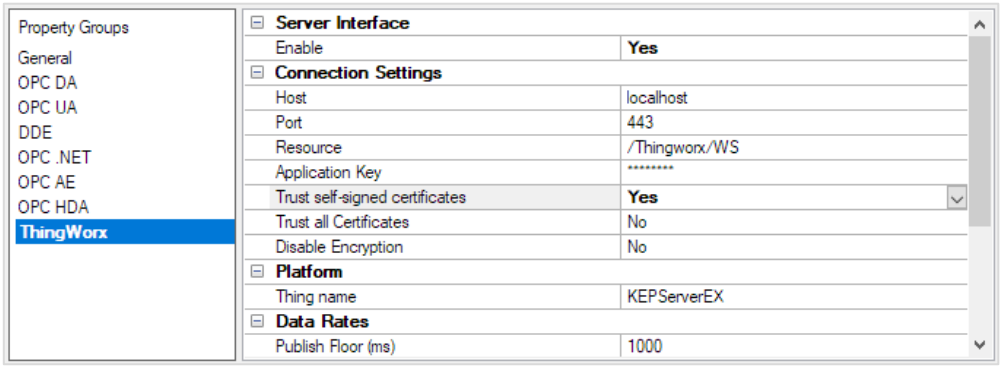
Secure AlwaysOn connections are possible with both self-signed certificates and Certificate Authority (CA) signed certificates. These instructions apply to Kepware products, such as KEPServerEX® and ThingWorx® Kepware® Server. For the remainder of this document, we will refer to only KEPServerEX.

1. Secure Connections using Self-Signed Certificates

1. Enable secure connections for self-signed certificates in ThingWorx,

For more information, see Configuring SSL/HTTPS with Self-Signed Certificate for the ThingWorx Platform on PTC eSupport.

2. Navigate to **Project Properties | ThingWorx** in KEPServerEX and configure Trust self-signed certificates to **Yes**.



2. Secure Connections with CA Signed Certificates

1. Enable secure connections for CA signed certificates in ThingWorx.

- *For more information, see [Configuring ThingWorx / Tomcat for SSL using a Signed Certificate from a Certificate Authority (CA)] on PTC eSupport.*

2. Obtain a copy of a CA signed certificate.
   - *See also [Obtain Certificate Authority Certificate].*
3. Access the Administration menu for KEPServerEX and select **Settings... | Certificate Store.**
   - **Note**: In the Certificate Store, ensure the correct product is selected from the dropdown menu.
4. Under **Extend Trust Store**, click **Import** to import the CA certificate.

No additional changes to the default KEPServerEX settings are required to ensure a secure connection with ThingWorx.

3. Obtain Certificate Authority Certificate

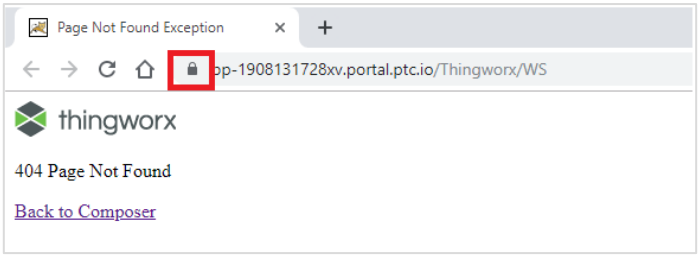
The following instructions use Google Chrome to make a copy of a CA certificate. However, any web browser should be able to copy a CA certificate.

1. In a web browser, navigate to the WebSocket URL of the ThingWorx instance:

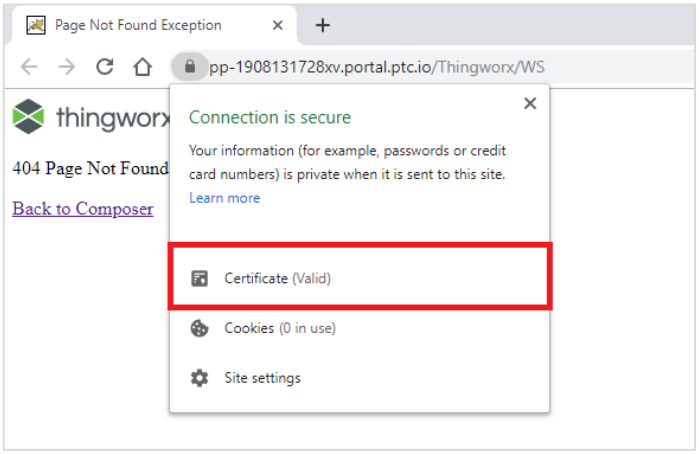
   ```
   <hostname>:<port>/Thingworx/WS
   ```

2. The browser should display a web server error and an HTML page that describes error "404 Page Not Found". This is expected.

3. Click the lock icon to the left of the URL in the Chrome address bar to access the site security information.

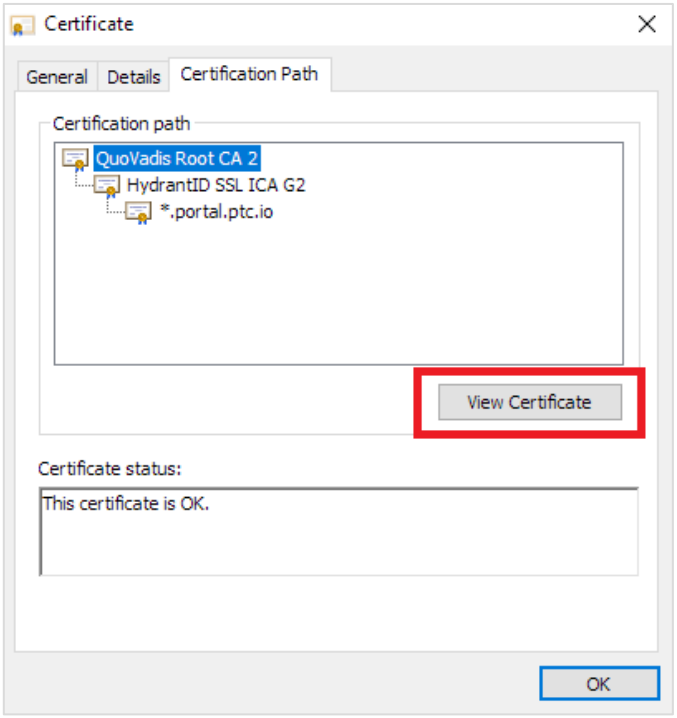


4. Select **Certificate** from the context menu.

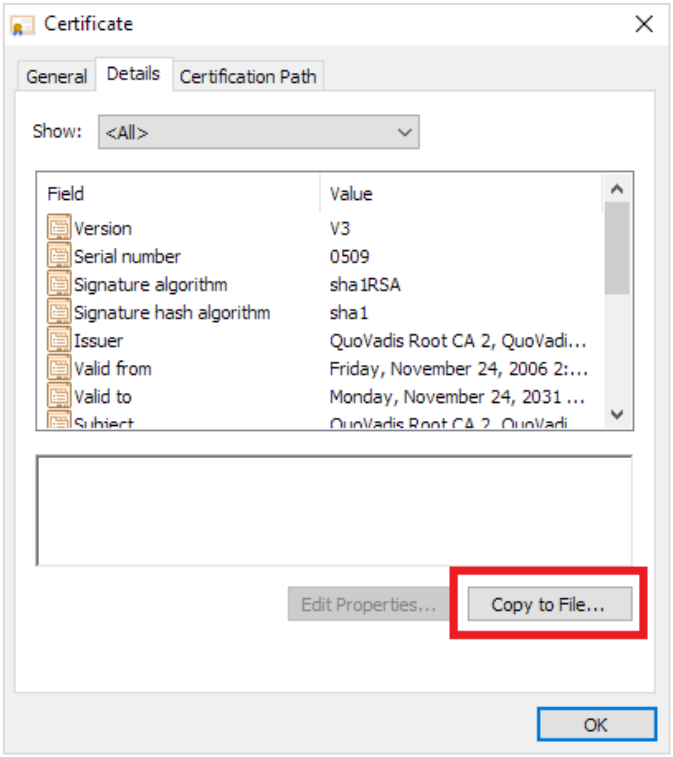


5. In the Certificate dialogue box, navigate to the Certification Path tab.

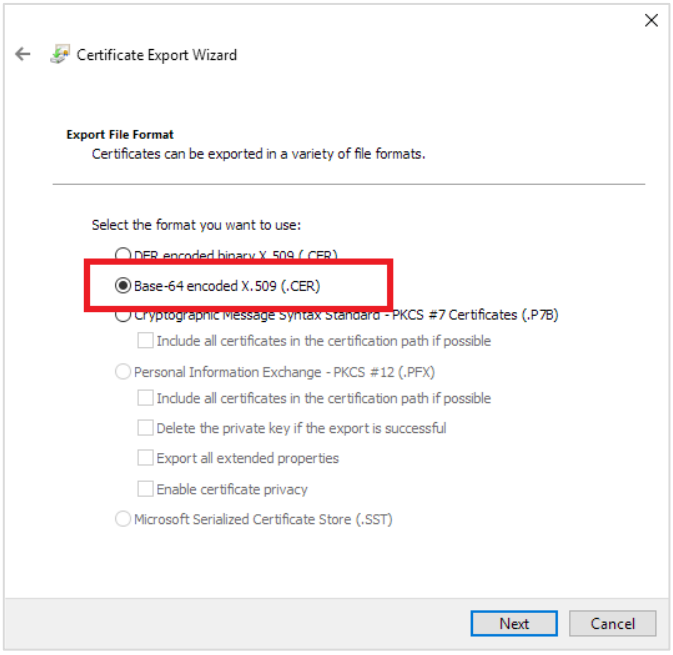
6. Select the top CA certificate from the Certification Path window and click **View Certificate**.



7. In the new Certificate dialogue box, navigate to the Details tab and click **Copy to File** to launch the Certificate Export Wizard.



8. Click **Next** in the Certificate Export Wizard, then select **Base-64 encoded X.509 (.CER)** for the export file format.



9. Specify a target directory and name for the certificate file and follow the remaining prompts to export the certificate.

4. Glossary

Certificate: A digital document that binds specific information to a public key so the receiver can confirm the identity of the sender.

Key: A specific string of numbers and letters that is applied by an algorithm to either encrypt or decrypt a message.

Public Key: A key that encrypts messages and can be made available to any party. A public key is also used to verify the signature of a private key. Messages encrypted with a given public key are decrypted with the corresponding private key.

Private Key: A key used in combination with a certificate and a public key to encrypt communication and verify identity. Any entity with access to a private key can impersonate the true owner of the key. Thus, private keys associated with certificates should be carefully guarded such that they cannot be stolen by malicious users.

x509 Certificate: A specific type of security certificate that includes information on the signature algorithm, Public Key, Serial Number, validation dates and an extended section which allows for additional custom fields. X509 Certificates are the standard certificate used in OPC UA and TLS-based communications.

Certificate Authority (CA): A third party that validates and signs public keys. The CA also maintains a list of certificates that have been revoked and should no longer be trusted. Think of a certificate authority as a place to get a passport. In order to receive an approved and signed document that proves an identity (passport), various forms of ID must be presented to the party issuing the passport. Therefore, an official passport is considered a valid form of identification. This same process applies to security certificates and public keys.

Self-Signed Certificate: A certificate that uses its own private key to sign itself. Think of a self-signed certificate as creating a passport and not getting it validated by the government. From a cybersecurity perspective, Self-Signed Certificates fall into two categories:

1. The entity doing the validation is the same entity creating the self-signed certificate (think of this as "personally" knowing the person creating the passport)
2. The entity doing the validation trusts every certificate by default, so the self-signed certificate is used only for data encryption, and not to establish trust.

Trust Store: A cache of trusted certificates or certificate authorities used to establish if an entity is trusted.