OPC .NET Configuration Manager

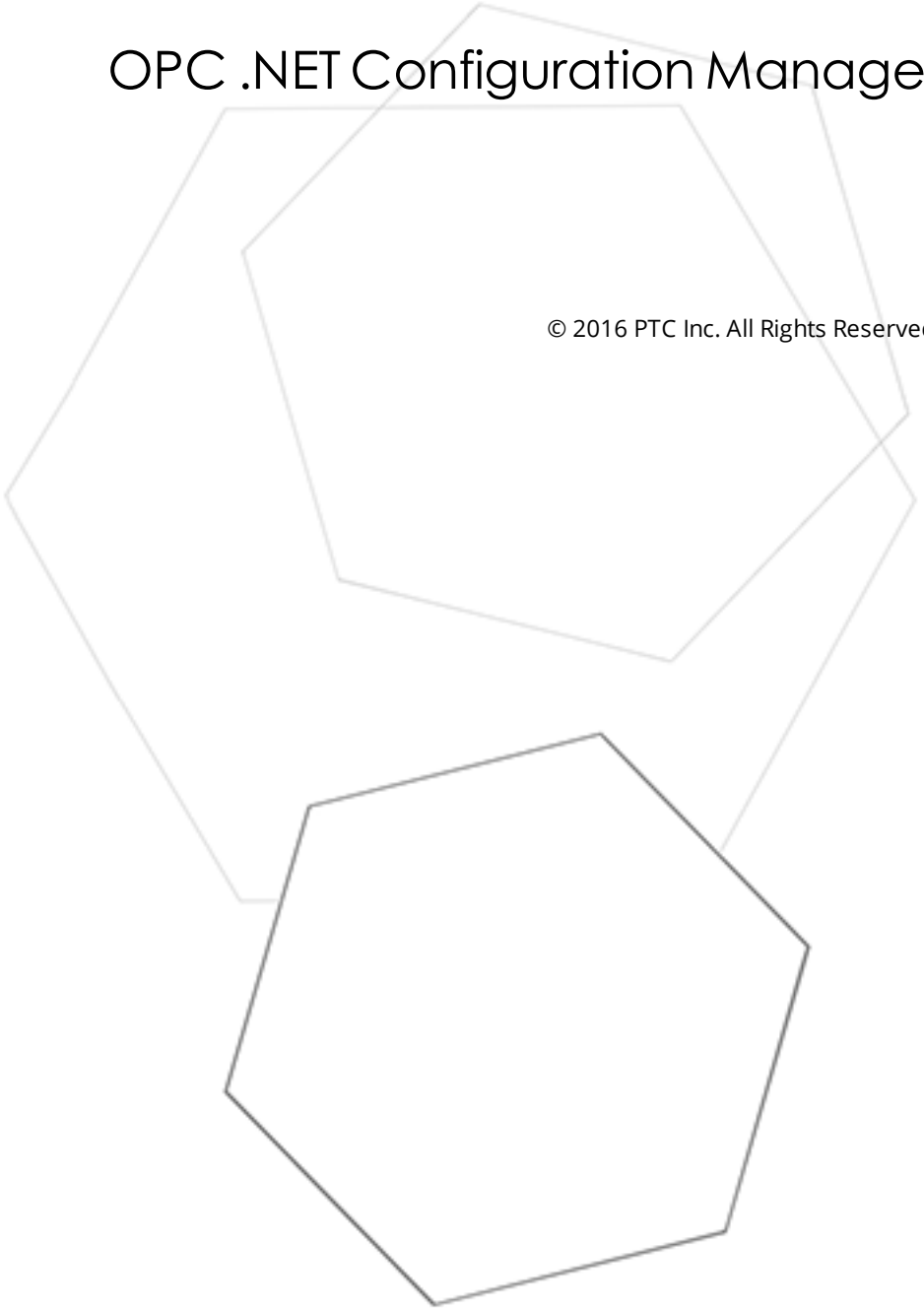


Table of Contents

OPC .NET Configuration Manager

Help version 1.016

CONTENTS

Overview

OPC .NET is a family of APIs provided by the OPC Foundation that leverage Microsoft's .NET technology and allow .NET clients to connect to the server.

The server supports OPC .NET 3.0 WCF, formally known as OPC Xi. Unlike other OPC .NET APIs, OPC .NET 3.0 uses Windows Communication Foundation (WCF) for connectivity, thus avoiding DCOM issues and providing the following benefits:

1. Secure communication via multiple communications bindings (such as Named Pipe, TCP, Basic HTTP, and Ws HTTP).
2. Consolidation of OPC Classic interfaces.
3. Simple development, configuration, and deployment of Windows environments.

The server adds OPC .NET 3.0 support using a customized version of the OPC .NET 3.0 WCF Wrapper supplied by the OPC Foundation. The wrapper runs as a service and wraps the existing server's OPC AE and DA interfaces. This provides WCF clients access to the server's tag and alarm data.

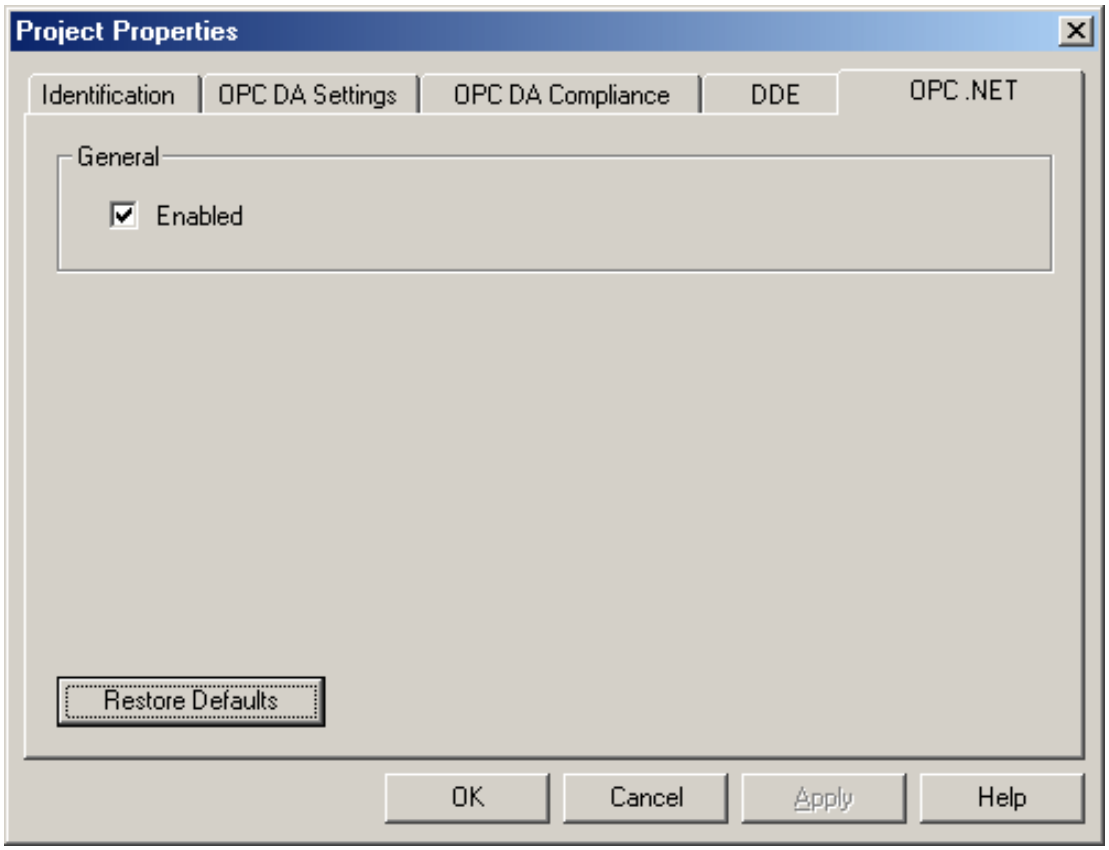
See Also: External Dependencies

External Dependencies

This application has external dependencies. OPC .NET 3.0 requires the installation of Microsoft's .NET 3.5 framework. If the framework is not installed prior to server installation, the wrapper and the OPC .NET Configuration Manager will not be available.

Server Settings

To access the OPC .NET server settings through the Configuration, click **File** | **Project Properties** and then select the **OPC .NET** tab.



Descriptions of the parameters are as follows:

- **Enable:** When checked, the OPC .NET Wrapper will be initialized and accept client connections.

Note: The OPC .NET Wrapper runs as a System Service called "xi_server_runtime.exe". It will only be started when the server starts and the option described above is enabled. Unlike OPC DA, clients cannot launch the server.

OPC .NET Configuration Manager

The OPC .NET Configuration Manager assists users in configuring the OPC .NET Wrapper settings (such as available communications bindings, wrapped server interfaces, and security settings). To access the OPC .NET Configuration Manager through the server Administration menu, right-click on the server icon in the System Tray and then select **OPC .NET Configuration**.

For more information on a specific OPC .NET Configuration Manager tab, select a link from the list below.

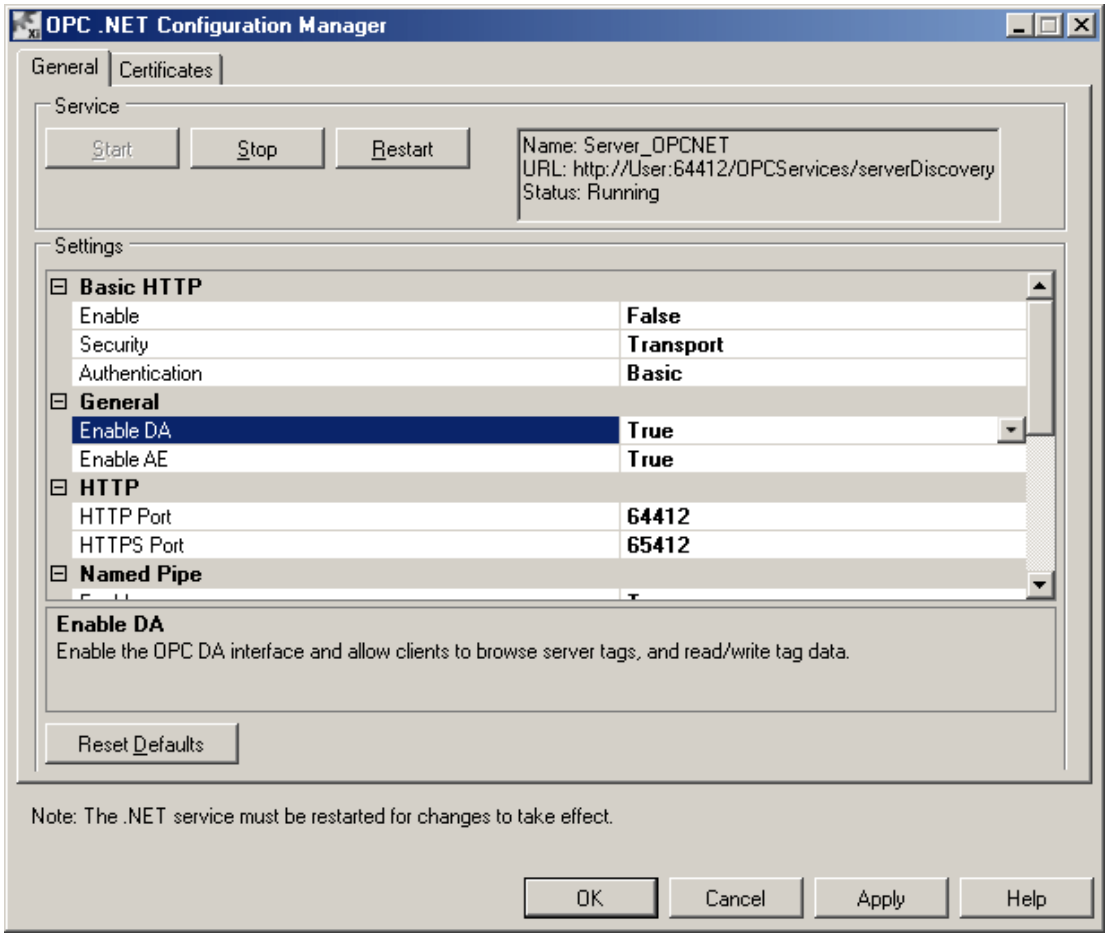
General
Certificates

Note: Some OPC .NET Configuration Manager functions require administrative privileges, such as certificate creation and management. These functions will not succeed without adequate privileges.

General

The General tab contains service and functionality controls.

Note: Users must restart the OPC .NET Wrapper in order for changes to take effect.



Service

Descriptions of the options are as follows:

- **Start:** This button starts the OPC .NET Wrapper service, thus allowing clients to connect.
- **Stop:** This button stops the OPC .NET Wrapper service, thus disconnecting clients.
- **Restart:** This button stops and restarts the OPC .NET Wrapper service. Users must restart the service in order for changes to take effect.

Note: If the service fails to start, refer to **Error Reporting** for more information.

Settings

Basic HTTP

Basic HTTP is a text-based protocol used for local and remote client/server communication by WS-I Basic Profile 1.1 compliant clients. Although it is generally used by older clients, the Ws HTTP binding is preferred. HTTP is slower than TCP, but is usually more interoperable. Descriptions of the options are as follows:

- **Enable:** When true, local or remote OPC .NET clients can connect using Basic HTTP.
- **Security:** This parameter controls the security used for client connections. For description of the security levels, refer to the table below.

Level	Description
None	No security will be used. Although this is the fastest option, it should only be used in a secure environment.
Transport	Transport security will be used, and provide endpoint to endpoint security. This security uses HTTPS, which requires the HTTPS port be configured in addition to a server SSL certificate.
Transport and Message	Transport and message security will be used. Although this is the most secure option, it is also the slowest. This security uses HTTPS, which requires the HTTPS port be configured in addition to a server SSL certificate.

- **Authentication:** This parameter controls how clients authenticate with the server. For description of the modes, refer to the table below.

Mode	Description
Basic	Client authentication is performed using a user name and password that must map to a valid Windows account name and password on the server.*
Certificate	Requires both the client and server to have and swap instance certificates. The server must trust the client certificate, and the client must trust the server certificate.

*If Basic authentication is used and Security is set to None, the client credentials will be sent as plain text.

See Also: **Connectivity Requirements**

General

The General settings control the OPC Classic interfaces that the OPC .NET Wrapper exposes to OPC .NET clients. Descriptions of the options are as follows:

- **Enable DA:** When true, this exposes the OPC Data Access (DA) interface to OPC .NET clients.
- **Enable AE:** When true, this exposes the OPC Alarms and Events (AE) interface to OPC .NET clients.

HTTP Ports
The HTTP Ports settings include secure HTTP (HTTPS), which is used for SSL communication. Descriptions of the options are as follows:

- **HTTP Port:** This port is used during server discovery, and is required for clients to browse the server. It is also used by the Basic HTTP and Ws HTTP bindings when the security modes are set to None. This port should not be used by any other application on the machine. If the machine running the server is using a firewall, the firewall must grant access to this port in order for clients to connect.
- **HTTPS Port:** This port is used by the Basic and Ws HTTP bindings when the security modes are set to Transport or Transport And Message. It is used for SSL communication, and should not be used by any other application on the machine. If the machine running the server is using a firewall, the firewall must grant access to this port in order for clients to connect.

Named Pipe
The Named Pipe binding is optimized for local client/server communication. Descriptions of the options are as follows:

- **Enable:** When true, local OPC .NET clients can connect using the Named Pipe binding.
- **Security:** This parameter controls the security used by the binding. For description of the security levels, refer to the table below:

Level	Description
None	No security will be used. Although this is the fastest option, it should only be used in a secure environment.
Transport	Transport security will be used, and provide endpoint to endpoint security.

TCP Binding
TCP Binding is used for fast local or remote binary communications. This binding is faster than HTTP bindings, but is usually less interoperable. Descriptions of the options are as follows:

- **Enable:** When true, local or remote OPC .NET clients will be able to connect using TCP.
- **Port:** This parameter specifies the port used by the TCP binding. It should not be used by any other application on the machine. If the machine running the server is using a firewall, the firewall must grant access to this port in order for clients to connect.
- **Security:** This parameter controls the security used for client connections. For description of the security levels, refer to the table below.

Level	Description
None	No security will be used. Although this is the fastest option, it should only be used in a secure environment.
Transport	Transport security will be used, and provide endpoint to endpoint security.
Transport and Message	Transport and message security will be used. Although this is the most secure option, it is also the slowest. This security uses SSL over TCP, and requires the client to trust the server instance certificate.

- **Authentication:** This parameter controls how clients authenticate with the server. For description of the modes, refer to the table below.

Mode	Description
Certificate	Requires both the client and server to have and swap instance certificates. The server must trust the client certificate, and the client must trust the server certificate.
Windows	Client authentication is performed using Windows domain credentials. This requires that the client and server be on the same domain.

See Also: Connectivity Requirements

Ws HTTP

Ws HTTP is a text-based binding used by local or remote clients for secure, reliable, non-duplex communication. It is generally used by newer clients, and is preferred over Basic HTTP. Although HTTP is slower than TCP, it is usually more interoperable. Descriptions of the options are as follows:

- **Enable:** When enabled, local or remote OPC .NET clients can connect using Ws HTTP.
- **Security:** This parameter controls the security used for client connection. For description of the security levels, refer to the table below.

Level	Description
None	No security will be used. Although this is the fastest option, it should only be used in a secure environment.
Transport	Transport security will be used, and provide endpoint to endpoint security. This security uses HTTPS, which requires that the HTTPS port be configured in addition to a server SSL certificate.
Transport and Message	Transport and message security will be used. Although this is the most secure option, it is also the slowest. This security uses HTTPS, which requires that the HTTPS port be configured in addition to a server SSL certificate.

- **Authentication:** This parameter controls how clients authenticate with the server. For descriptions of the modes, refer to the table below.

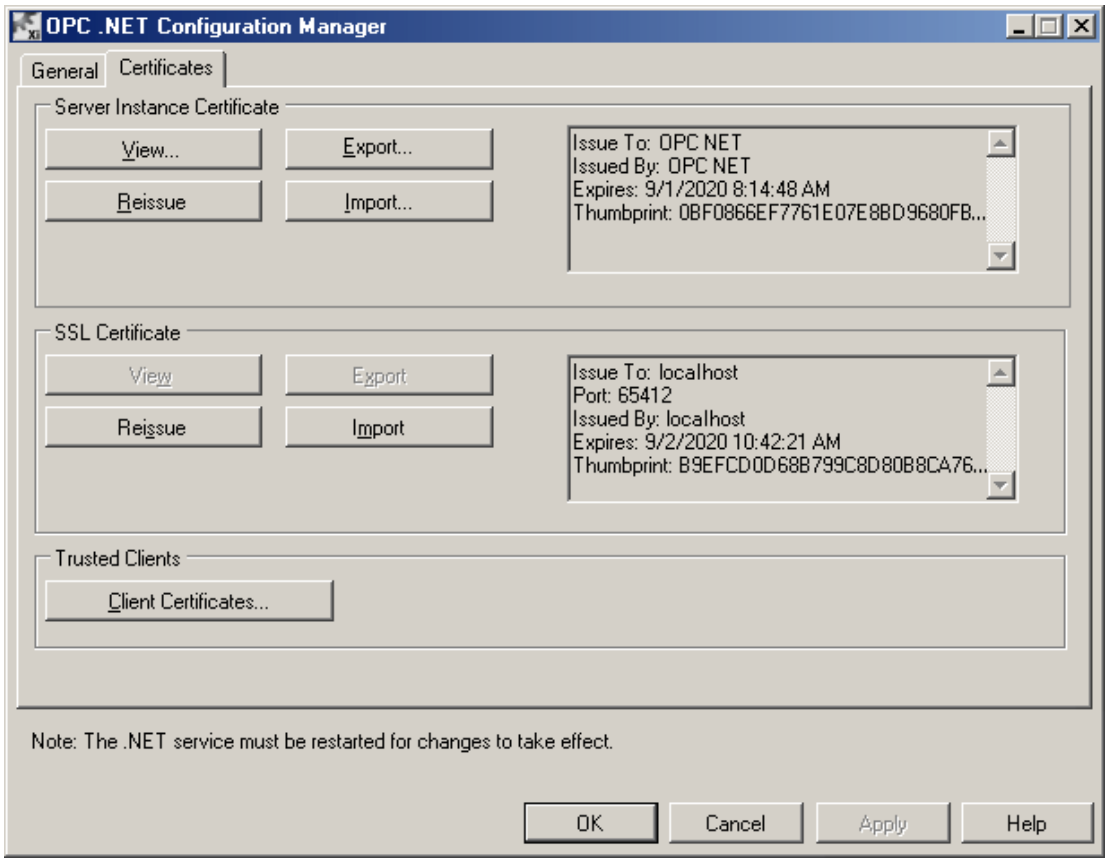
Mode	Description
Basic	Client authentication is performed using a user name and password that must map to a valid Windows account name and password on the server. Basic authentication is not available when the Security setting is None.
Certificate	Requires both the client and server to have and swap instance certificates. The server must trust the client certificate, and the client must trust the server certificate.
Windows	Client authentication is performed using Windows domain credentials. This requires that the client and server be on the same domain.

- **Restore Defaults:** This option restores all properties to their default settings. For this change to take effect, users must click **Apply** or **OK**.

See Also: Connectivity Requirements

Certificates

The Certificates tab specifies the configuration for the server instance certificate, server SSL certificates, and trusted client certificates.



Service Instance Certificate

The Server Instance Certificate uniquely identifies the server, and is used for certificate authentication and SSL over TCP. In both cases, the client must trust the server certificate in order to establish connectivity.

Descriptions of the options are as follows:

- **View...:** When clicked, this will display the server instance certificate using the Windows Certificate Viewer. A subset of the information will be displayed on this page.
- **Export...:** When clicked, this will export the server instance certificate's public key, allowing clients to import the certificate and establish a trust relationship.
- **Import...:** When clicked, this will import a certificate to use as the server instance certificate. This allows use of certificates that have been signed by a Certificate Authority (CA). Imported certificates must be in .pfx file format, and may require a password for import.

  Note: This operation requires that the OPC .NET Configuration Manager run under an administrative account.

- **Reissue:** When clicked, this will reissue the certificate using a self-signed certificate. Self-signed certificates cannot be verified by a CA, and are not usually used in production environments. Although they are ideal for testing, it is recommended that users import a CA signed certificate to use as the server instance certificate.

  Note: This operation requires that the OPC .NET Configuration Manager run under an administrative account.

Server SSL Certificate

An SSL certificate is required when the Basic or Ws HTTP bindings are configured to use Transport or Transport And Message security. The SSL certificate is only used for SSL, and should not be confused with the server instance certificate (which is used for authentication and secure TCP).

The SSL certificate's Common Name (CN) is required to match an address of the host machine. For example, if the server is hosted on a machine with an IP of 10.10.220.10 and wants clients to connect using SSL over this IP, it must issue a certificate with a Common Name equal to 10.10.220.10. Clients must trust the SSL certificate, and then connect to the server using the 10.10.220.10 address. It is required that the CN match the host address in order to verify that the certificate is coming from the machine to which the client is connecting.

Descriptions of the parameters are as follows:

- **View:** When clicked, this will display the SSL certificate using the Windows Certificate Viewer. A subset of the information will be displayed on the page.
- **Export:** When clicked, this will export the SSL certificate's public key, allowing clients to import the certificate and establish a trust relationship.
- **Import:** When clicked, this will import a certificate for use as the SSL certificate. A certificate that has been signed by a Certificate Authority (CA) can be used as the SSL certificate. Imported certificates must be in .pfx file format, and may require a password.

  Note: For this operation, the OPC .NET Configuration Manager must be run under an administrative account.
- **Reissue:** When clicked, this will reissue the certificate using a self-signed certificate. Self-signed certificates cannot be verified by a CA, and are not usually used in production environments. Although they are ideal for testing, it is recommended that users import a CA signed certificate to use as the server instance certificate.

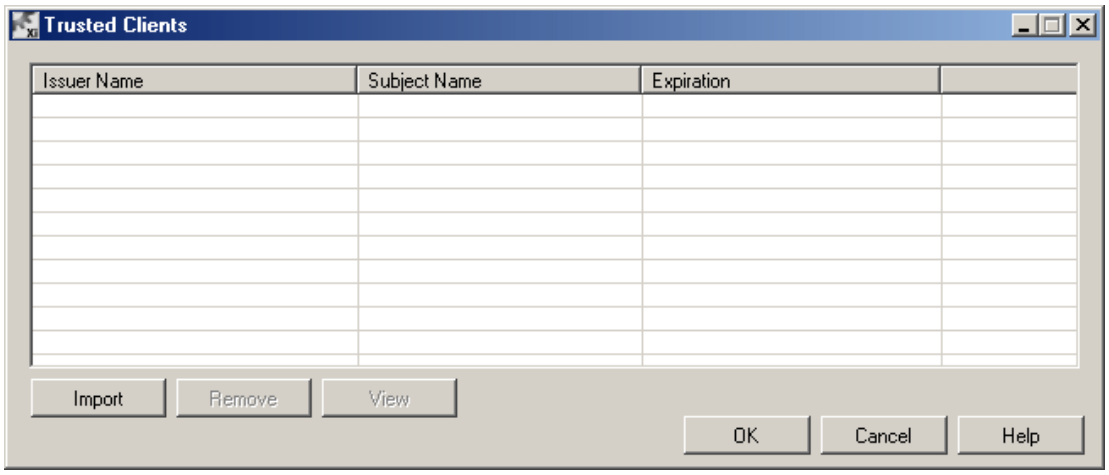
  In **Host Name**, either select the host name (such as "localhost" or the machine name) or enter one manually. The certificate's Common Name must be a valid host name or IP for the server.

  Note: For this operation, the OPC .NET Configuration Manager must be run under an administrative account.

Important: The Reissue and Import operations require both administrative privileges and access to either the Windows "netsh.exe" utility or the "httpcfg.exe" utility. External Microsoft utilities are used to map the SSL certificate to the specific IP and port on the host machine. On Windows Vista and up, "netsh.exe" is used and is installed by default. On Windows XP and below, "httpcfg.exe" is used and is not installed by default. It can be downloaded from Microsoft's website at **Windows XP Service Pack 2 Support Tools**. Users must perform a complete installation in order to get the utility, and should reboot the machine after the install completes.

Trusted Clients

When clicked, Client Certificates will launch the Trusted Clients dialog, which is used to define trusted certificates. Trust relationships are required for certificate authentication.

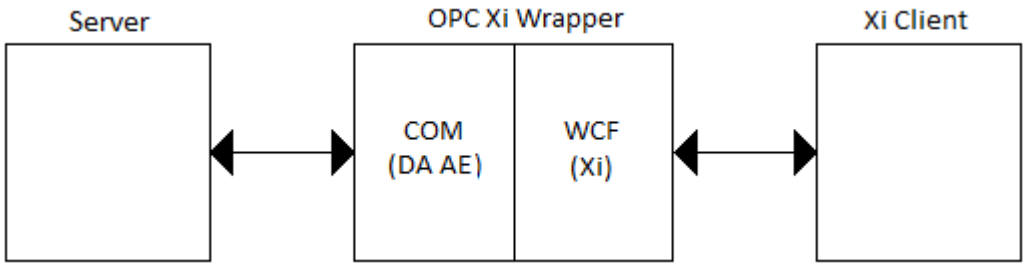


Descriptions of the options are as follows:

- **Import:** When clicked, this button will import a trusted client. The client certificate is placed in the Windows certificate store under Local Machine, Trusted People. This operation requires administrative privileges. For more information, refer to **Managing Certificates in Windows**.
- **Remove:** When clicked, this button will remove a client certificate from the certificate store. This operation requires administrative privileges.
- **View:** When clicked, this button will display a client certificate using the Windows Certificate Viewer.

OPC .NET Wrapper

The OPC .NET Wrapper wraps the server's OPC Data Access (DA) and OPC Alarms and Events (AE) interfaces. It runs as a manual service, which starts and stops with the primary server. As such, OPC .NET clients cannot launch the server like Classic OPC clients.



For more information on a specific part of the OPC .NET Wrapper, select a link from the list below.

OPC .NET Supported Interfaces
OPC .NET Server Discovery
Error Reporting

OPC .NET Supported Interfaces

The OPC .NET Wrapper supports the following interfaces.

Interface	Description
IResourceManagement	OPC .NET clients' entrance point. This interface manages client connections, the browsing of the server namespace, and clients' creation and management of item lists used for reading and writing server data.
IRead	This interface reads tag and alarm data from the server.
IWrite	This interface writes tags and acknowledges alarms in the server.
ICallback	This interface supports an asynchronous callback that is used to notify clients of server data changes and events. It is not supported by the Basic and Ws HTTP bindings because it requires full duplex communication.
IRegisterForCallback	This interface is used by clients to register for the ICallback interface.
IPoll	This interface allows clients to poll for data changes and events, rather than receive them asynchronously. The interface is generally used by bindings that don't support ICallback or clients that want more control over the reception of data change notifications.
IServerDiscovery	This interface allows clients to locate OPC .NET servers and their IResourceManagement endpoints.*

*For more information on discovery, refer to **OPC .NET Server Discovery**.

Note: Although the IRestRead interface is not officially supported, it can be manually configured. For more information, refer to **Advanced Configuration**.

Supported Interfaces

OPC .NET hosts endpoints, which consist of an interface and a binding. Because of binding limitations, not all bindings support all interfaces. The table below displays the interfaces that are supported by each binding.

Interface	Named Pipe	TCP	Basic Http	Ws Http
IResourceManagement	X	X	X	X
IRead	X	X	X	X
IWrite	X	X	X	X
ICallback	X	X		
IRegisterForCallback	X	X		
IPoll	X	X	X	X
IServerDiscovery			X	

OPC .NET Server Discovery

OPC .NET Server Discovery allows OPC .NET clients to automatically discover OPC .NET servers on the network. The method of discovery is specific to each client. In most cases, an OPC .NET Server Discovery application is run side-by-side with the OPC .NET client. The application can discover OPC .NET servers in one of three ways:

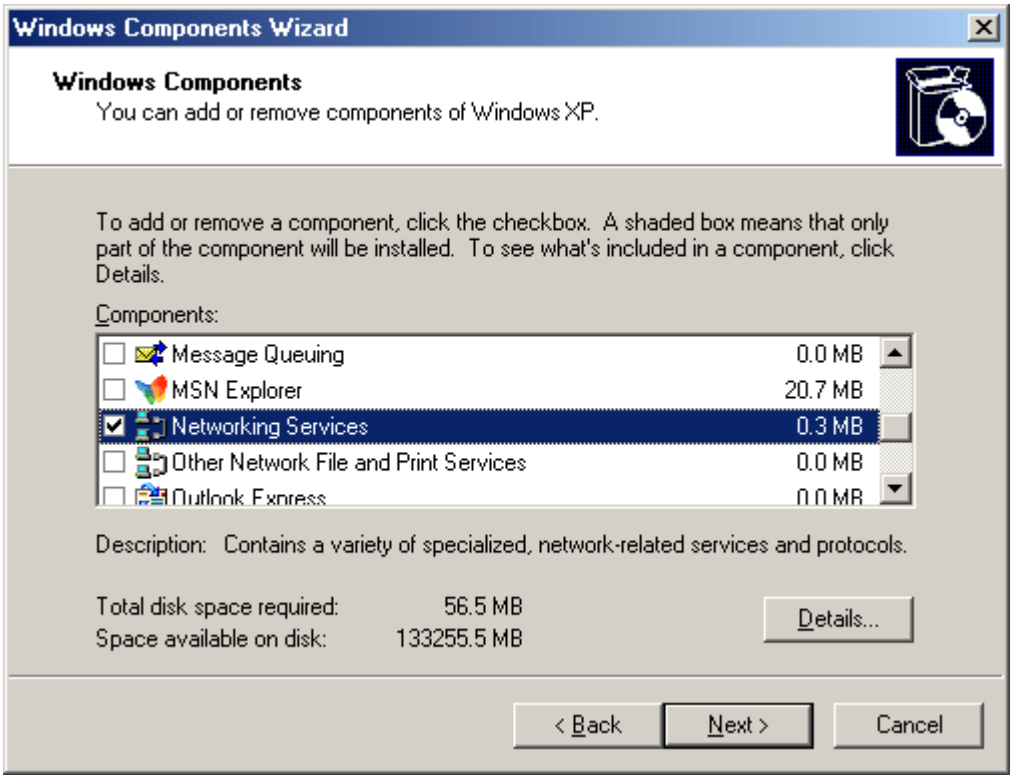
1. By explicitly specifying a server's discovery URL. This does not require server configuration.
2. By allowing users to manually enter discovery URLs. This does not require server configuration.
3. Through use of Microsoft's Peer Name Resolution Protocol (PNRP). This requires that PNRP be enabled on the server machine.

PNRP on Windows XP and Windows Server 2003

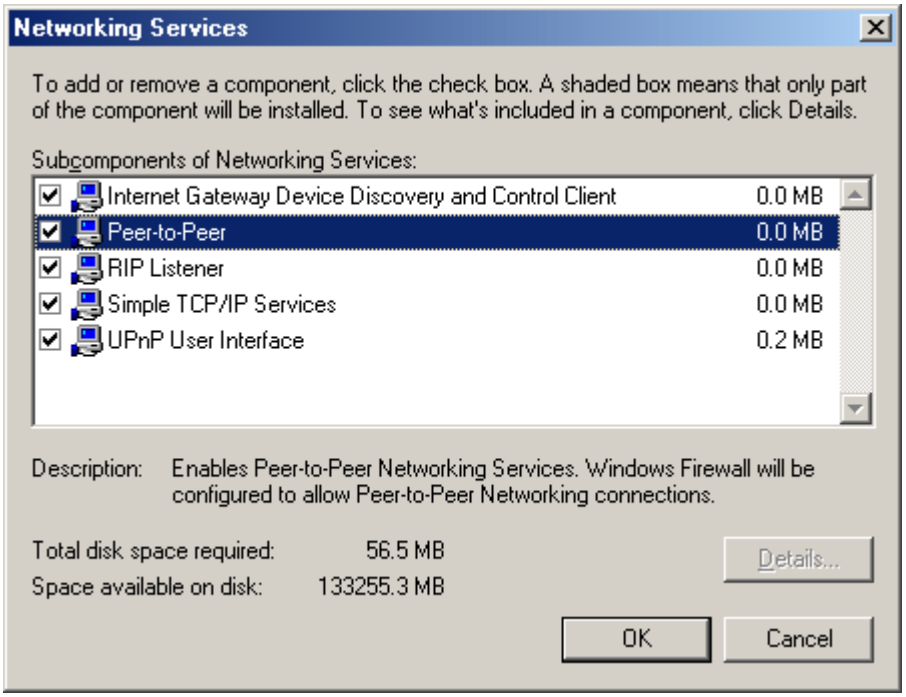
PNRP is supported on SP2 and up, but is not installed by default. For information on installing PNRP, refer to the instructions below.

1. To start, navigate to the **Control Panel**. Then, open **Add or Remove Programs**.
2. Next, select **Add/Remove Windows Components**.

3. Check **Networking Services**, and then click **Details**.



4. Check **Peer-to-Peer**, and then click **OK**.



5. Click **Next** to start the the PNRP service installation.

6. Next, open the **Windows Service Control Manager**. Then, start the **Peer Name Resolution Protocol** service.

Note: PNRP discovery is now enabled.

PNRP on Windows 7, Vista, and Windows Server 2008

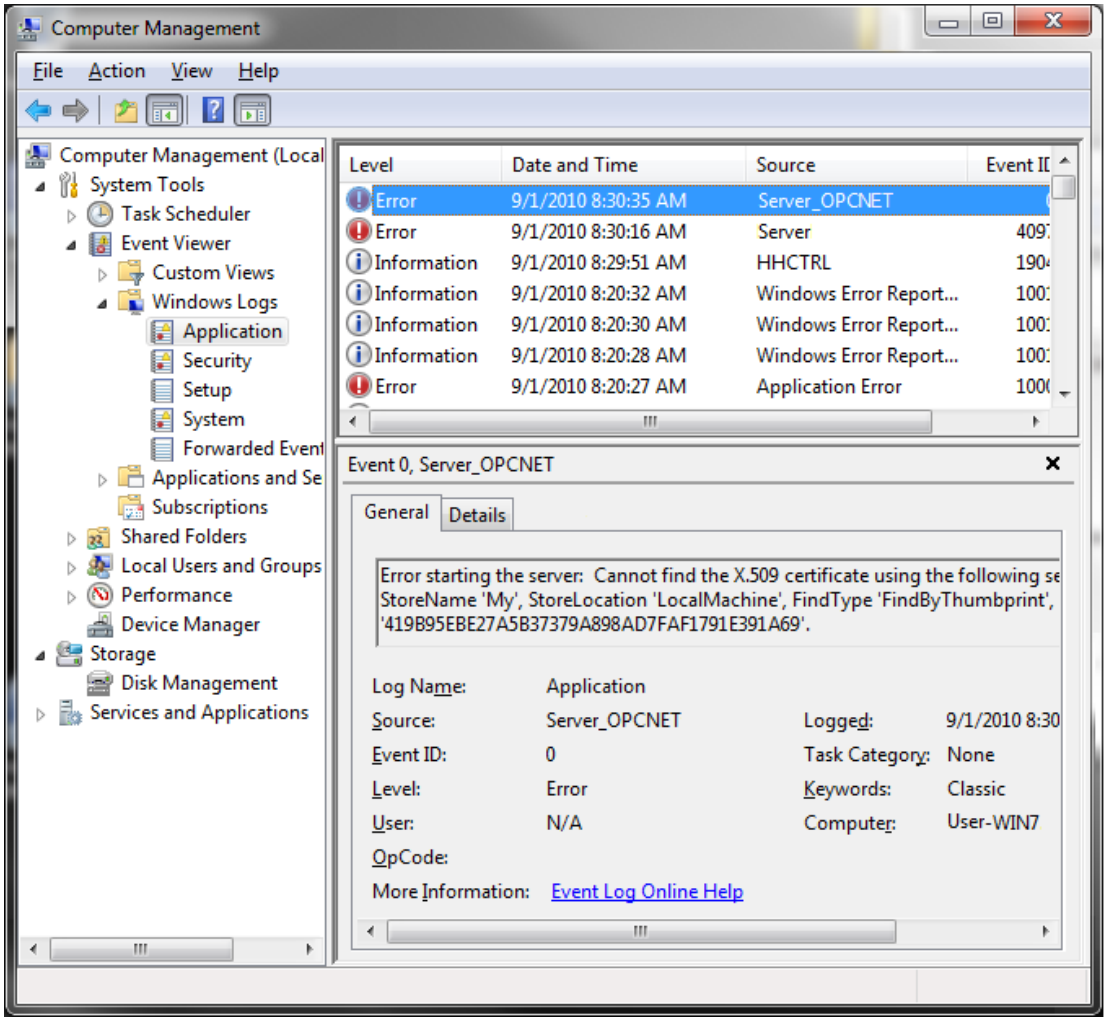
PNRP is fully supported. It requires that the Peer Name Resolution Protocol and Peer Networking Identity Manager services be running.

Error Reporting

The OPC .NET Wrapper service reports errors using the Windows Event Log. If the OPC .NET Wrapper fails to start, launch the Windows Event Log by doing the following:

1. First, open the **Start** menu.
2. Right-click on **My Computer** and then select **Manage**. This will invoke the **Computer Management** console.
3. Next, click **System Tools** | **Event Viewer** | **Windows Logs** | **Application**. The Event Log messages will be displayed in the top middle pane.

   Note: The source of OPC .NET Wrapper error events is the OPC .NET Wrapper service name.
4. Click on an event to display the details.



Connectivity Requirements

Security Modes

The server's connectivity requirements for the different binding security modes and authentication types are discussed in detail below. Use the provided information as a guide for setting up secure OPC .NET connections.

Note: Clients are responsible for their own configuration.

Binding	Security	Description
Named Pipe	None	Transport security is used by the IResourceManagement endpoint. The protection level is EncryptAndSign. All other endpoints have no security. No setup is required.
	Transport and Message	Transport security is used by all interfaces. The protection level is EncryptAndSign. No setup is required.
TCP	None	Transport security is used by the IResourceManagement endpoint. The protection level is EncryptAndSign. All other endpoints have no security. No setup is required.
	Transport	Transport security is used for all endpoints. The protection level is EncryptAndSign. No setup is required.
	Transport and Message	Transport and message security for all endpoints (TransportWithMessageCredential). The protection level is EncryptAndSign. This configuration requires the server to have a valid server instance certificate, which must be trusted by the OPC .NET client.
Basic Http	None	The IResourceManagement endpoint uses TransportCredentialOnly security. No setup is required.*
	Transport	Transport security for all endpoints. This configuration requires SSL.**
	Transport and Message	Transport and message security for all endpoints (TransportWithMessageCredential). This configuration requires SSL.**
Ws Http	None	The IResourceManagement endpoint uses Message security. All other endpoints have no security. No setup is required.
	Transport	Transport security for all endpoints. This configuration requires SSL.**
	Transport and Message	Transport and message security for all endpoints (TransportWithMessageCredential). This configuration requires SSL.**

*If authentication is set to Basic, the client credentials will be sent in plain text.
**For more information, refer to "SSL Requirements" below.

SSL Requirements

SSL requires that the server has a valid SSL certificate whose Subject Common Name (CN) matches the IP of the client that is connecting. This is important: if the certificate's CN is 10.10.200.10 but the client connects to localhost, the connection will fail. The client must connect to 10.10.200.10. It must also trust the certificate.

Authentication Types

The table below details each authentication type's connectivity requirements, and also discusses the server configuration.

Note: Clients are responsible for their own configuration.

Authentication	Description
Certificate	Certificate authentication requires that both the client and server have valid instance certificates, and that they trust each other's certificates. Users must first import the client certificate using the OPC .NET Configuration Manager's Trusted Clients dialog. Then, the server will trust the client certificate. Next, users must export the server's instance certificate, and configure the client to trust it. For more information, refer to "Certificate Exception" below.
Windows	Windows authentication requires that both the client and server be on the same domain. The server must have access the domain server in order to validate the client's credentials.*
Basic	Basic authentication requires that the client provide a user name and password. The credentials must be linked to a Windows account on the server. For example, if a client provides "John Doe" and "123," there must be a "John Doe Windows" account on the server with the password "123."

*For most OPC .NET clients, the client credentials are the logged-in user's Windows credentials.

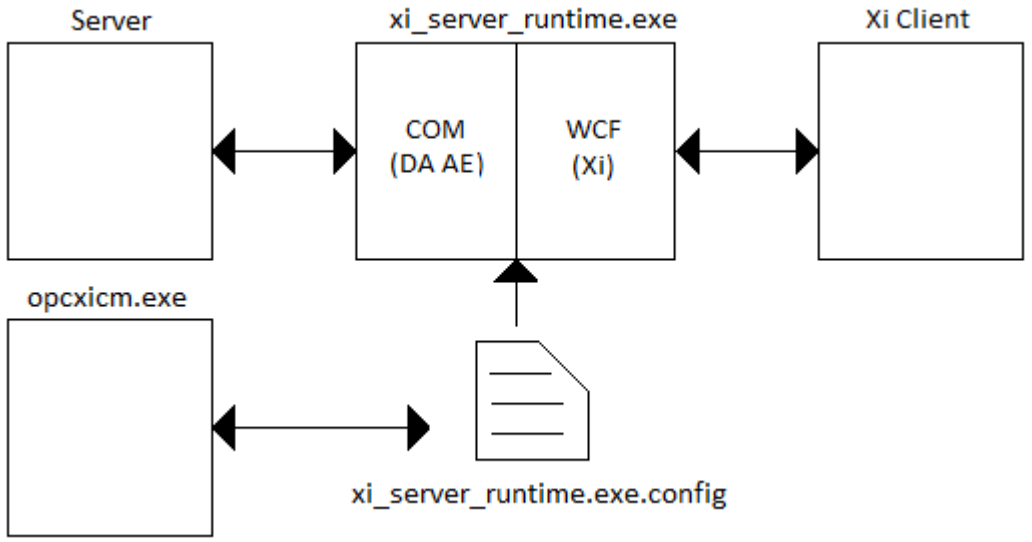
Certificate Exception

If the server is hosted on Windows XP, the client instance certificate will be self-signed. If the server is using the Basic or Ws HTTP binding with Transport security, importing the client certificate via the OPC .NET Configuration Manager may not work. The OPC .NET Configuration Manager places the certificate in the Local Machine's Trusted People store. In this case, the certificate must be traceable to a trusted root. Because the certificate is self-signed, it must be placed in the Local Machine's Trusted Root Certificate Authorities store. For more information, refer to **Managing Certificates in Windows**.

Advanced Configuration

WCF Configuration

OPC .NET is configured via an "xi_server_runtime.exe.config" file, which is located in the product's shared application directory beneath "Xi". The OPC .NET Configuration Manager saves simple WCF configuration information to this file, which is then loaded by the OPC .NET Wrapper. The wrapper uses the configuration to host endpoints.



When enabled, the binding will expose a single endpoint on each supported interface. The supported interfaces are displayed in the table below. Each 'X' represents an endpoint.

Note: Only the configurations listed below are supported, and some clients may require additional configuration. To account for this, administrators can edit the "xi_server_runtime.exe.config" file directly. This documentation does not discuss WCF configuration, as it is assumed that advanced users have a working knowledge of WCF.

Interface	NetNamedPipe	NetTCP	Basic Http	Ws Http
IResourceManagement	X	X	X	X
IRead	X	X	X	X
IWrite	X	X	X	X
ICallback	X	X		
IRegisterForCallback	X	X		
IPoll	X	X	X	X
IServerDiscovery			X	

Custom Configuration

The "xi_server_runtime.exe.config" file contains a custom "<appSettings>" section. The only settings in this section that are required by the OPC .NET Wrapper service are "enableda" and "enableae". These settings are used to enable the wrapping of the server's OPC DA and AE interfaces. If it is not present, the interfaces will not be wrapped.

```
<appSettings>
  <add key="enableda" value="True" />
  <add key="enableae" value="True" />
```

</appSettings>

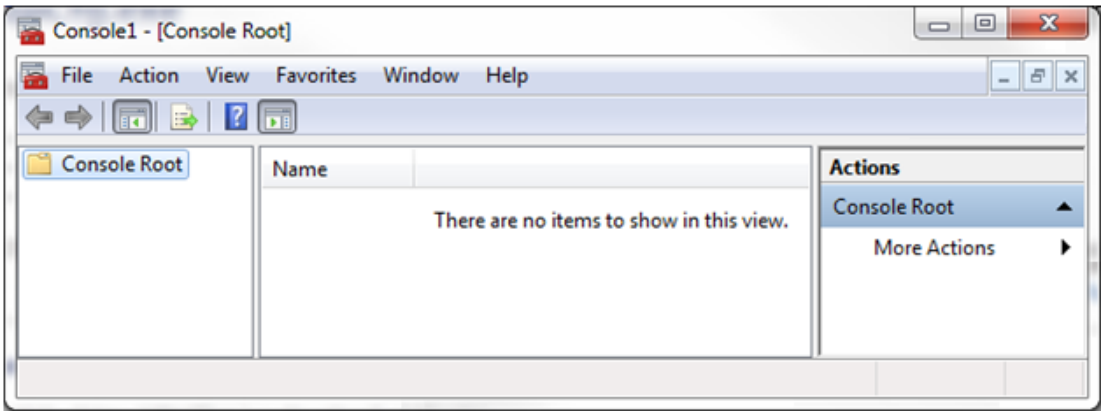
Note: Users can edit the content in the <system.serviceModel> section in order to add custom endpoints or bindings.

Caution: Using the OPC .NET Configuration Manager after editing the configuration file will delete the custom configuration.

Managing Certificates in Windows

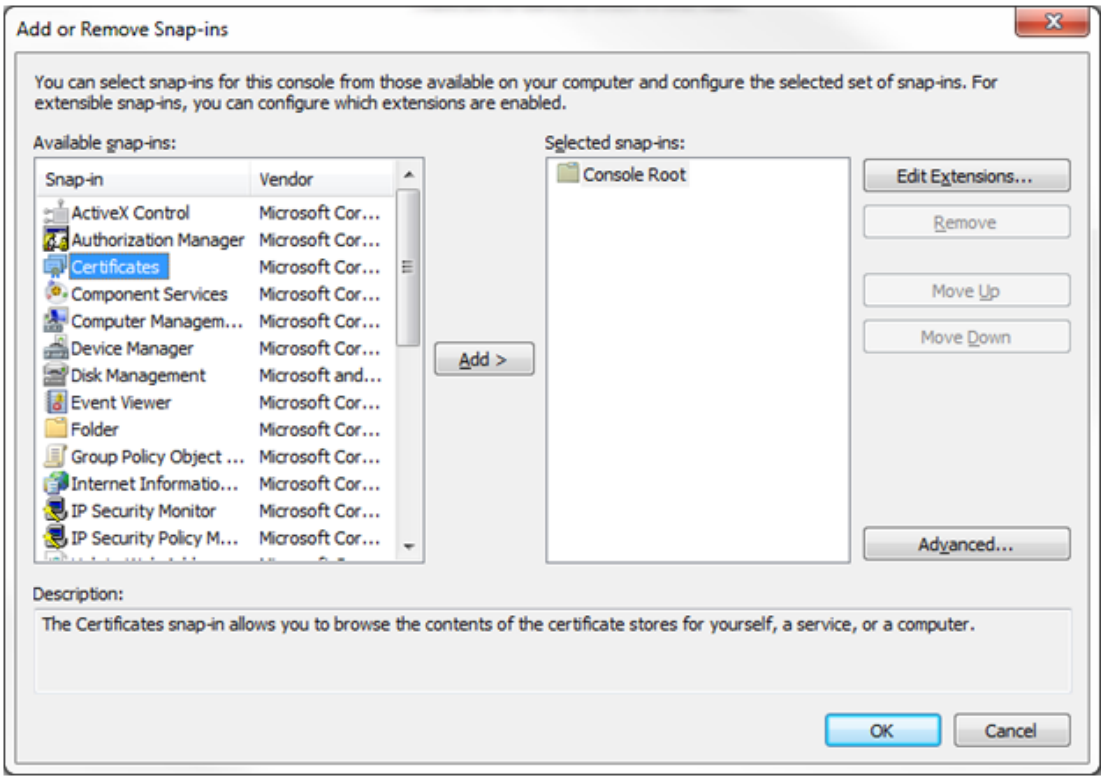
There have been cases where client applications do not provide certificate configuration support. As such, users will benefit by understanding Windows certificate management even though the OPC .NET Configuration Manager should be sufficient for managing certificates on the server. For more information, refer to the instructions below.

1. Click **Start** | **Run**. Then, specify "mmc.exe" to access Microsoft's certificate management tool.

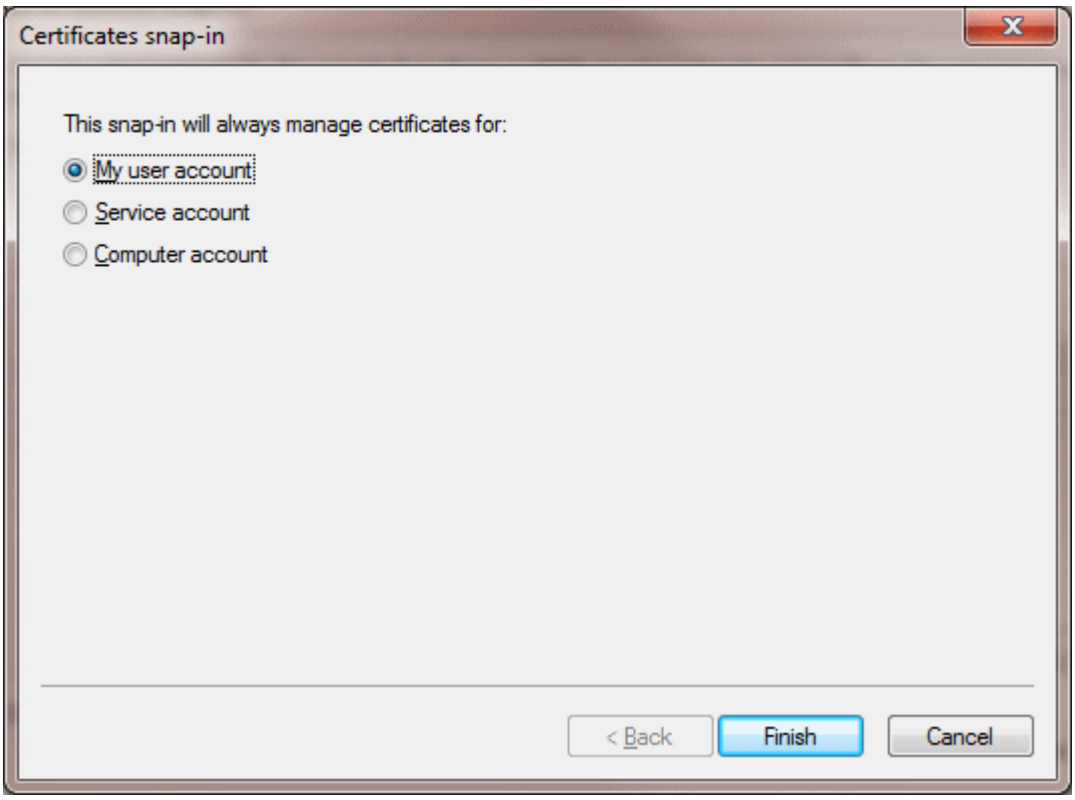


2. Next, click **File** | **Add/Remove Snap-in...**.

3. Beneath **Available Snap-ins**, select **Certificates**.



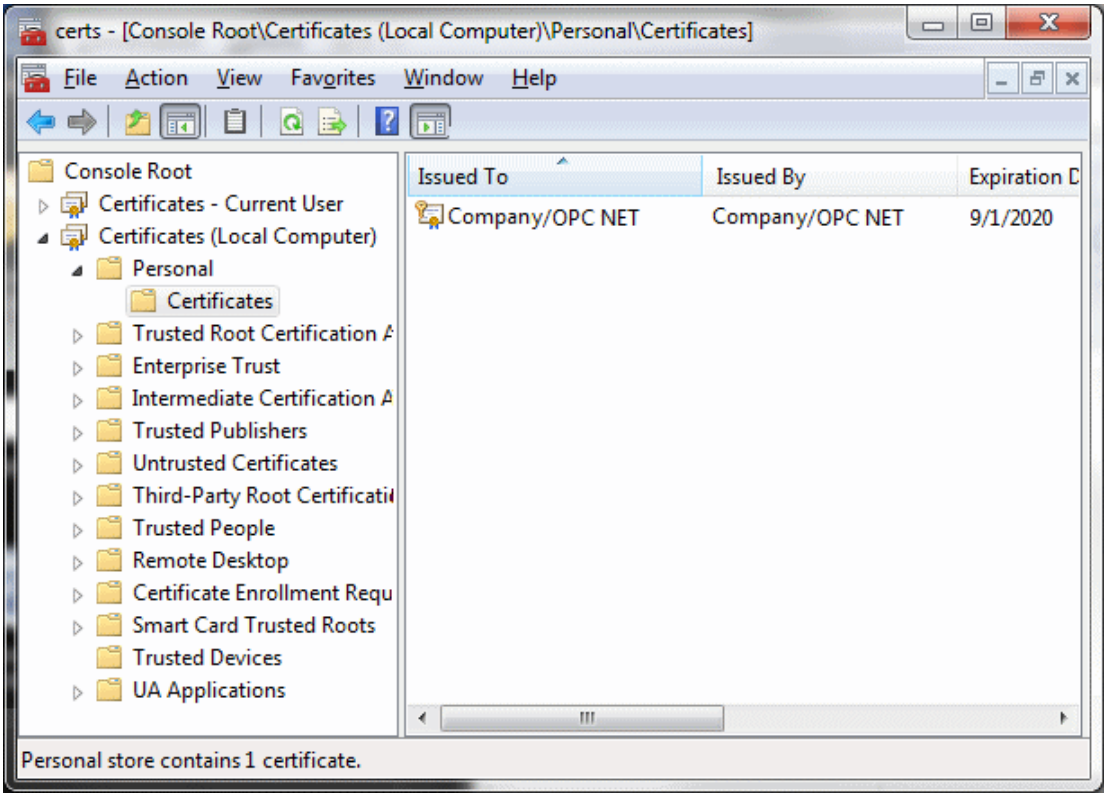
4. Next, click **Add >**. There are three certificate locations available for selection.



Descriptions of the accounts are as follows:

- **My User Account:** This account is the local logged-in user. It is typically used by OPC .NET clients.
- **Service Account:** This account is linked to a local or remote service.
- **Computer Account:** This account is also called "Local Machine," and is machine-wide.

5. Select **Computer Account** and **My User Account** and then click **Finish**.



6. Next, click **File** | **Save** to save the configuration. This will allow users to load it when running the "mmc.exe" tool again.
7. Microsoft manages certificates in multiple stores. Options include Personal, Trusted People, and Trusted Root Certificate Authority. Descriptions of the stores are as follows:

- **Personal Store:** This store holds certificates for locally installed applications; both the OPC .NET Wrapper instance certificate and the SSL certificates are stored in the Personal store. In this store, users can view, import, and export the OPC .NET Wrapper certificate (which is the same certificate that is visible from the Certificate tab in the OPC .NET Configuration Manager).
- **Trusted People:** This store holds the certificates of local or remote clients that the OPC .NET Wrapper trusts. These certificates are required for certificate authentication. Users can right-click on the store and then import trusted client certificates, which will invoke the same Trusted Clients dialog that is located in the OPC .NET Configuration Manager.
- **Trusted Root Certificate Authority:** This store holds the Certificate Authorities (CA) that are trusted by the Trusted People store. With this setup, a certificate chain is built to verify

the validity of the client certificate.

Note: If the server is running on Windows XP, and the Basic or Ws HTTP bindings are used with Transport security and certificate authentication, self-signed client certificates must be placed in the Trusted Root Certificate Authority.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

The OPC .NET Wrapper failed to start because it is not installed. Please rerun the installation

Error Type:

Error

Possible Cause:

The OPC .NET Wrapper is not installed.

Solution:

Install the OPC .NET Wrapper.

The OPC .NET Wrapper failed to start. Please see the windows application event log for more details. Also make sure the .NET 3.5 Framework is installed

Error Type:

Error

Possible Cause:

The server failed to start the "xi_server_runtime.exe" service. This could occur if there is a WCF configuration error, an OPC COM error, or if the service is not set to manual start.

Solution:

1. If an error message is present in the Windows Event Log, the error is a result of an invalid WCF configuration. Information for correcting the error should be provided in the message. Otherwise, connect to the server using an OPC DA or an OPC AE client.
2. If an error message is not present in the Windows Event Log, check the Windows Service Control Manager and make sure that the OPC .NET service is installed and set to manual startup. Otherwise, connect to the server using an OPC DA or an OPC AE client.
3. Reinitialize, restart, or reinstall the server.
4. Repair the Microsoft .Net installation.

Troubleshooting Tips

Select a link from the list below for more information.

Unable to start the OPC .NET Wrapper service

Possible Cause:

1. The OPC .NET Instance Certificate is not installed.
2. The OPC .NET service is not installed.
3. The OPC .NET Wrapper has an invalid configuration.
4. A timeout occurred.

Solution:

1. Make sure that the OPC .NET Instance Certificate is installed and is valid by checking the OPC .NET Configuration Manager Certificates. If the certificate is invalid, either reissue it or import a new one.
2. Make sure that the service is installed by checking the OPC .NET Configuration Manager General tab. If the service is not installed, re-run the installation.
3. Check the Windows Event Log for more information on the error. To fix the configuration, re-run the OPC .NET Configuration Manager and then select "Restore Defaults" (located in the General tab).
4. The OPC .NET service took too long to start (5+ seconds). This is generally not an error: the OPC .NET service will eventually start.

Unable to make changes to the OPC .NET certificates

Possible Cause:

The user does not have administrative privileges on the machine.

Solution:

The OPC .NET Wrapper uses the Windows certificate store to store certificates. Write access to the store requires administrative privileges. Log in with an administrator account to make changes to the certificates.

Unable to change the SSL certificate

Possible Cause:

1. "netsh.exe" or "httpcfg.exe" is not installed on the machine. These utilities are required to map the certificate to the Windows HTTP stack.
2. The user does not have administrative privileges on the machine.

Solution:

1. For information on installing the "netsh.exe" and "httpcfg.exe" utilities, refer to **OPC .NET Configuration Manager**.
2. The OPC .NET Wrapper uses the Windows certificate store to store certificates. Write access to the store requires administrative privileges. Log in with an administrator account to make changes to the certificates.

Index

S

T

U