

OPC UA .NET Standard Installation

Installation and Administration of .NET Standard based
OPC UA Applications





Document Control

Version	Date	Comment
1.0.8	18-MAY-2019	Initial version based on SDK 1.0.8
1.0.9	31-MAY-2019	Enhanced Manual Installation chapter and added the Prerequisites chapter
1.1.0	10-JUN-2019	Enhanced OPC UA Security and Configuration Tool chapters
1.1.1	26-JUL-2019	Updated to new evaluation downloads
1.2.0	11-OCT-2019	<ul style="list-style-type: none">- Added informationen for .NET Core 2.0 on Linux, macOS- Changed to .NET 4.6.2- Removed OPC UA Client Gateway (no longer supported)- Removed Sample Binary Installer (no longer supported)

Purpose and audience of document

This document describes how to deploy and administer OPC UA Applications from Technosoftware GmbH and applications build on either the OPC UA Client SDK .NET Standard or the OPC UA Server SDK .NET Standard. The target audience for this document are systems administrators.



Referenced OPC Documents

Documents	
This document partly uses extracts taken from the OPC UA specifications to be able to give at least a short introduction into the specifications. The specifications itself are available from: http://www.opcfoundation.org/Default.aspx/01_about/UA.asp?MID=AboutOPC#Specifications	
OPC Unified Architecture Textbook, written by Wolfgang Mahnke, Stefan-Helmut Leitner and Matthias Damm: http://www.amazon.com/OPC-Unified-Architecture-Wolfgang-Mahnke/dp/3540688986/ref=sr_1_1?ie=UTF8&s=books&qid=1209506074&sr=8-1	
[OPC 10000-1]	OPC UA Specification: Part 1 – Overview and Concepts https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/
[OPC 10000-2]	OPC UA Specification: Part 2 – Security Model https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-2-security-model/
[OPC 10000-3]	OPC UA Specification: Part 3 – Address Space Model https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-3-address-space-model/
[OPC 10000-4]	OPC UA Specification: Part 4 – Services https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-4-services/
[OPC 10000-5]	OPC UA Specification: Part 5 – Information Model https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-5-information-model/
[OPC 10000-6]	OPC UA Specification: Part 6 – Mappings https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-6-mappings/
[OPC 10000-7]	OPC UA Specification: Part 7 – Profiles https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-7-profiles/
[OPC 10000-8]	OPC UA Specification: Part 8 – Data Access https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-8-data-access/
[OPC 10000-9]	OPC UA Specification: Part 9 – Alarm & Conditions https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-9-alarms-and-conditions/
[OPC 10000-10]	OPC UA Specification: Part 10 – Programs https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-10-programs/
[OPC 10000-11]	OPC UA Specification: Part 11 – Historical Access https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-11-historical-access/
[OPC 10000-12]	OPC UA Specification: Part 12 – Discovery and Global Services https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-12-discovery-and-global-services/
[OPC 10000-13]	OPC UA Specification: Part 13 – Aggregates https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-13-aggregates/
[OPC 10000-14]	OPC UA Specification: Part 14 – PubSub https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-14-pubsub/
[OPC 10000-100]	OPC UA Specification Part 100 – Devices https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-100-device-information-model/



Other Referenced Documents

SOAP Part 1: SOAP Version 1.2 Part 1: Messaging Framework

<http://www.w3.org/TR/soap12-part1/>

SOAP Part 2: SOAP Version 1.2 Part 2: Adjuncts

<http://www.w3.org/TR/soap12-part2/>

XML Encryption: XML Encryption Syntax and Processing

<http://www.w3.org/TR/xmlenc-core/>

XML Signature: XML-Signature Syntax and Processing

<http://www.w3.org/TR/xmldsig-core/>

WS Security: SOAP Message Security 1.1

<http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>

WS Addressing: Web Services Addressing (WS-Addressing)

<http://www.w3.org/Submission/ws-addressing/>

WS Trust: Web Services Trust Language (WS-Trust)

<http://specs.xmlsoap.org/ws/2005/02/trust/WS-Trust.pdf>

WS Secure Conversation: Web Services Secure Conversation Language (WS-SecureConversation)

<http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf>

SSL/TLS: RFC 2246: The TLS Protocol Version 1.0

<http://www.ietf.org/rfc/rfc2246.txt>

X200 : ITU-T X.200 – Open Systems Interconnection – Basic Reference Model

<http://www.itu.int/rec/T-REC-X.200-199407-I/en>

:X509: X.509 Public Key Certificate Infrastructure

<http://www.itu.int/rec/T-REC-X.509-200003-I/e>

HTTP: RFC 2616: Hypertext Transfer Protocol - HTTP/1.1

<http://www.ietf.org/rfc/rfc2616.txt>

HTTPS: RFC 2818: HTTP Over TLS

<http://www.ietf.org/rfc/rfc2818.txt>

IS Glossary: Internet Security Glossary

<http://www.ietf.org/rfc/rfc2828.txt>

NIST 800-12: Introduction to Computer Security

<http://csrc.nist.gov/publications/nistpubs/800-12/>

NIST 800-57: Part 3: Application-Specific Key Management Guidance

http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_PART3_key-management_Dec2009.pdf

NERC CIP: CIP 002-1 through CIP 009-1, by North-American Electric Reliability Council

<http://www.nerc.com/page.php?cid=2|20>

IEC 62351: Data and Communications Security

http://www.iec.ch/heb/d_mdock-e050507.htm



SPP-ICS: System Protection Profile

Industrial Control System, by Process Control Security Requirements Forum (PCSRF)

<http://www.isd.mel.nist.gov/projects/processcontrol/SPP-ICSv1.0.pdf>

SHA-1: Secure Hash Algorithm RFC

<http://tools.ietf.org/html/rfc3174>

PKI: Public Key Infrastructure article in Wikipedia

http://en.wikipedia.org/wiki/Public_key_infrastructure

X509 PKI: Internet X.509 Public Key Infrastructure

<http://www.ietf.org/rfc/rfc3280.txt>

EEMUA : 2nd Edition EEMUA 191 - Alarm System - A guide to design, management and procurement (Appendixes 6, 7, 8, 9).

<http://www.eemua.co.uk/>



TABLE OF CONTENTS

1	Installation .NET Core 2.0	9
2	Installation .NET 4.6.2 / .NET 4.7.2	10
2.1	Directory Structure	11
2.2	Prerequisites	12
2.3	DLL's used by applications based on the SDK	12
2.3.1	.NET 4.6.2 and .NET 4.7.2	12
2.3.2	.NET Standard 2.0	13
2.4	OPC UA Local Discovery Server	13
2.5	OPC UA Configuration Tool	14
2.6	OPC UA Sample Server	15
2.7	OPC UA Sample Client	16
3	OPC UA Sample Server and OPC UA Sample Client Usage	17
4	OPC UA Security	21
4.1	Background	21
4.2	Security Tiers	23
4.2.1	The Basics	23
4.2.2	Tier 1 - No Authentication	23
4.2.3	Tier 2 - Server Authentication	23
4.2.4	Tier 3 - Client Authentication	24
4.2.5	Tier 4 - Mutual Authentication	24
4.3	Certificates and Certificate Stores	25
4.3.1	Overview	25
4.3.2	Certificates and Private Keys	25
4.3.3	Windows Certificate Stores	26
4.3.4	Directory Stores	27
4.3.5	X509 Stores	28
4.4	Key Certificate Properties	29
5	Tools	30
5.1	OPC UA Configuration Tool	30
5.1.1	Overview	30
5.1.2	Choosing an Application to Manage	31
5.1.3	Manage Application	32
5.1.4	Manage Security	35



5.1.5	Manage Application Certificate	37
5.1.6	Manage Certificates	42
5.2	Microsoft Management Console	44



Disclaimer

© Technosoftware GmbH. All rights reserved. No part of this document may be altered, reproduced or distributed in any form without the expressed written permission of Technosoftware GmbH.

This document was created strictly for information purposes. No guarantee, contractual specification or condition shall be derived from this document unless agreed to in writing. Technosoftware GmbH reserves the right to make changes in the products and services described in this document at any time without notice and this document does not represent a commitment on the part of Technosoftware GmbH in the future.

While Technosoftware GmbH uses reasonable efforts to ensure that the information and materials contained in this document are current and accurate, Technosoftware GmbH makes no representations or warranties as to the accuracy, reliability or completeness of the information, text, graphics, or other items contained in the document. Technosoftware GmbH expressly disclaims liability for any errors or omissions in the materials contained in the document and would welcome feedback as to any possible errors or inaccuracies contained herein.

Technosoftware GmbH shall not be liable for any special, indirect, incidental, or consequential damages, including without limitation, lost revenues or lost profits, which may result from the use of these materials. All offers are non-binding and without obligation unless agreed to in writing.

Trademark Notice

Microsoft, MSN, Windows and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.



1 Installation .NET Core 2.0

The OPC UA Client & Server SDK .NET Standard can be used not only on Windows but also on Linux and macOS. The SDKs includes a client and server application targeting .NET Core 2.0. To be able to use them you need to install .NET Core on your system.

Please follow instructions in this [article](#) to setup the dotnet command line environment for your platform. As of today, .Net Standard 2.0 is required. The article describes the installation of .NET SDK 2.2.401 for Windows, Linux and macOS. This version also works with the OPC UA Client and Server Solutions we provide.

Please follow at least the sections

- [Intro](#)
- [Download and Install](#)

to install the .NET Core.

How to build and use the example applications provided is explained in the documents

- [OPC UA Client Development with NET Standard](#)
- [OPC UA Server Development_with NET Standard](#)



2 Installation .NET 4.6.2 / .NET 4.7.2

For starting with OPC UA Development you can download the **OPC UA Bundle SDK .NET Standard** from <https://technosoftware.com/evaluations/>.

The basic directory layout of the zip file content is as follows:

- **opc-ua-sdk-net-standard-v12.zip**
The OPC UA Client & Server SDK .NET Standard assemblies and examples
- **OPC_UA_NET_Standard_Installation_Guide.pdf**
This guide
- **OPC_UA_Client_Development_with_NET_Standard.pdf**
OPC UA Client Development tutorial based on the Workshop Clients
- **OPC_UA_Server_Development_with_NET_Standard.pdf**
OPC UA Server Development tutorial based on the Workshop Servers

Manual Installation requires the content of the opc-ua-sdk-net-standard-v12.zip zip file. So please unzip it and follow the next chapters to install the different applications.



2.1 Directory Structure

The zip file contains the following basic directory layout:

- **bin/**
 - **net462/**
Standard SDK Executables and DLL's for the .NET 4.6.2 Framework
 - **net472/**
Standard SDK Executables and DLL's for the .NET 4.7.2 Framework
 - **netstandard2.0/**
Standard SDK Executables and DLL's for the .NET Standard 2.0 and .NET Core 2.0 Framework
 - **redist/**
 - **OPC UA Local Discovery Server 1.03/**
The installer and Merge-Module for the OPC UA Local Discovery Server
- **doc/**
Additional documentation like the compiled HTML Help files for Server and Client SDK.
 - **pdf/**
Several documentation files. The more important ones here are:
 - **OPC_UA_NET_Standard_Installation_Guide.pdf**
This document
 - **OPC_UA_SDKs_NET_Standard_Introduction.pdf**
Introduction in Developing OPC UA Clients and OPC UA Servers with C# / VB.NET
 - **OPC_UA_Client_Development_with_NET_Standard.pdf**
Tutorial for Developing OPC UA Clients with C# / VB.NET based on the Workshop Client
 - **OPC_UA_Server_Development_with_NET_Standard.pdf**
Tutorial for Developing OPC UA Servers with C# / VB.NET based on the Workshop Server
- **examples/**
Sample applications
- **keys/**
The dummy Key for signing the executables and DLL's
- **schema/**
XSD files like the UAModelDesign.xsd used for the Model Designer
- **scripts/**
Scripts and executables used for building the applications
- **Workshop/**
Workshop presentation files as PDF



2.2 Prerequisites

The following prerequisites are required:

- **Visual C++ Redistributable for Visual Studio 2013**
It seems that on some installations the OPC UA Local Discovery Server installer doesn't install this prerequisite. It is only required for the OPC UA Local Discovery Server.

2.3 DLL's used by applications based on the SDK

The SDK consists of the following main components

- **Opc.Ua.Core.dll**
- **Technosoftware.UaConfiguration.dll**

These two DLL's are used by all applications using the SDK. In addition, one or several of the following DLL's might be required:

- **Technosoftware.UaClient.dll**
Client Applications require this DLL.
- **Technosoftware.UaServer.dll**
Server Applications require this DLL.

2.3.1 .NET 4.6.2 and .NET 4.7.2

The DLL's can be found in

- **bin/**
 - **net462/**
Standard SDK Executables and DLL's for the .NET 4.6.2 Framework
 - **net472/**
Standard SDK Executables and DLL's for the .NET 4.7.2 Framework

The main components require the following DLL's which can be found in the corresponding directories:

- BouncyCastle.Crypto.dll
- Newtonsoft.Json.dll
- Interop.NetFwTypeLib.dll



2.3.2 .NET Standard 2.0

DLL's used by applications can be found in

- **bin/**
 - **netstandard2.0/**
Standard SDK Executables and DLL's for the .NET Standard 2.0 and .NET Core 2.0 Framework

The main components require the following DLL's which can be found in the corresponding directory:

- BouncyCastle.Crypto.dll
- Newtonsoft.Json.dll

2.4 OPC UA Local Discovery Server

The Local Discovery Server (LDS) is a DiscoveryServer that maintains a list of all UA Servers and Gateways available on the host/PC that it runs on and is the UA equivalent to the OPC Classic OPCENUM interface.

An LDS is a service that runs in the background. UA Servers will periodically connect to the LDS and Register themselves as being available. This periodic activity means that the list of available UA servers is always current and means that a Client can immediately connect to any of them (security permissions pending).

The OPC UA Local Discovery Server is an installation from the OPC Foundation and delivered as installation executable and as merge module. These files are included in the OPC UA Client SDK .NET Standard and OPC UA Server SDK .NET Standard and can be found in the directory

\bin\redist\OPC UA Local Discovery Server 1.03

You can integrate the merge module OPC_UA_Local_Discovery_Server_1.03.msm into your application setup or use the OPC UA Local Discovery Server 1.03.400.431.exe for installation.

Important: The OPC UA Local Discovery Server doesn't work on Windows XP !

If you can't install the OPC UA Certificate Generator you should install the certificate generator as described in the next chapter. Otherwise you can skip that step.

The LDS also installs online documentation which is typically available at

[file:///C:/Program Files \(x86\)/Common Files/OPC Foundation/UA/Discovery/doc/index.htm](file:///C:/Program Files (x86)/Common Files/OPC Foundation/UA/Discovery/doc/index.htm)

Installation:

Start the OPC UA Local Discovery Server 1.03.400.431.exe and follow the instructions of the installer.

Important:

It seems that on some installations the OPC UA Local Discovery Server installer doesn't install the **Visual C++ Redistributable for Visual Studio 2013**. Since V1.0.9 our sample binary installer includes this as well.



2.5 OPC UA Configuration Tool

The OPC UA Configuration Tool is required for configuration of the OPC UA Applications like OPC UA Sample Server, OPC UA Sample Client and OPC UA Client Gateway or for certificate handling.

The following main components and its dependencies are required:

- Opc.Ua.Core.dll
- Technosoftware.UaConfiguration.dll
- Technosoftware.UaClient.dll
- Technosoftware.UaServer.dll
- BouncyCastle.Crypto.dll
- Newtonsoft.Json.dll
- Interop.NetFwTypeLib.dll

In addition to the main components mentioned above the following files are required:

- Technosoftware.ConfigurationTool.exe
- Technosoftware.ConfigurationTool.Config.xml
- Technosoftware.ConfigurationTool.exe.config
- Interop.ActiveDs.dll
- Technosoftware.CommonControls.dll

All files must be copied to the same directory.

Versions for .NET 4.6.2 and .NET 4.7.2 are delivered with the OPC UA Client SDK .NET Standard and the OPC UA Server SDK .NET Standard. It can be found at \bin\net462 or \bin\net472.

After applying the steps in the chapters before you must install the OPC UA Configuration Tool. This is possible by using the following command line parameters handled by the OPC UA Configuration Tool executable:

- | | |
|------------|--|
| /install | Installs and configures the OPC UA Configuration Tool. |
| /uninstall | Uninstalls the application by removing the changes made during installation. |

Installation:

Open a command line window, change directory to the location of the executable and use

Technosoftware.ConfigurationTool.exe /install



2.6 OPC UA Sample Server

The following main components and its dependencies are required to install the OPC UA Sample Server:

- Opc.Ua.Core.dll
- Technosoftware.UaConfiguration.dll
- Technosoftware.UaServer.dll
- BouncyCastle.Crypto.dll
- Newtonsoft.Json.dll
- Interop.NetFwTypeLib.dll

In addition to the main components mentioned above the following files are required:

- Technosoftware.SampleServer.exe
- Technosoftware.SampleServer.Config.xml
- Technosoftware.SampleServer.exe.config
- Technosoftware.CommonControls.dll
- Technosoftware.UaServer.Controls.dll
- Technosoftware.SampleControls.dll

All files must be copied to the same directory.

Versions for .NET 4.6.2 and .NET 4.7.2 are delivered with the OPC UA Client SDK .NET Standard and the OPC UA Server SDK .NET Standard. It can be found at \bin\net462 or \bin\net472.

After applying the steps in the chapters before you must install the OPC UA Sample Server. This is possible by using the following command line parameters handled by the OPC UA Sample Server executable:

- | | |
|------------|---|
| /install | Installs and configures the OPC UA Sample Server, e.g. certificates are created, and firewall configured. |
| /uninstall | Uninstalls the application by removing the changes made during installation. |

Installation:

Open a command line window, change directory to the location of the executable and use

```
Technosoftware.SampleServer.exe /install
```



2.7 OPC UA Sample Client

The following main components and its dependencies are required to install the OPC UA Sample Client:

- Opc.Ua.Core.dll
- Technosoftware.UaConfiguration.dll
- Technosoftware.UaClient.dll
- BouncyCastle.Crypto.dll
- Newtonsoft.Json.dll
- Interop.NetFwTypeLib.dll

In addition to the main components mentioned above the following files are required:

- Technosoftware.SampleClient.exe
- Technosoftware.SampleClient.Config.xml
- Technosoftware.SampleClient.exe.config
- Technosoftware.CommonControls.dll
- Technosoftware.UaClient.Controls.dll
- Technosoftware.SampleControls.dll

All files must be copied to the same directory.

Versions for .NET 4.6.2 and .NET 4.7.2 are delivered with the OPC UA Client SDK .NET Standard and the OPC UA Server SDK .NET Standard. It can be found at \bin\net462 or \bin\net472.

After applying the steps in the chapters before you must install the OPC UA Sample Client. This is possible by using the following command line parameters handled by the OPC UA Sample Server executable:

- | | |
|------------|---|
| /install | Installs and configures the OPC UA Sample Client, e.g. certificates are created, and firewall configured. |
| /uninstall | Uninstalls the application by removing the changes made during installation. |

Installation:

Open a command line window, change directory to the location of the executable and use

```
Technosoftware.SampleClient.exe /install
```

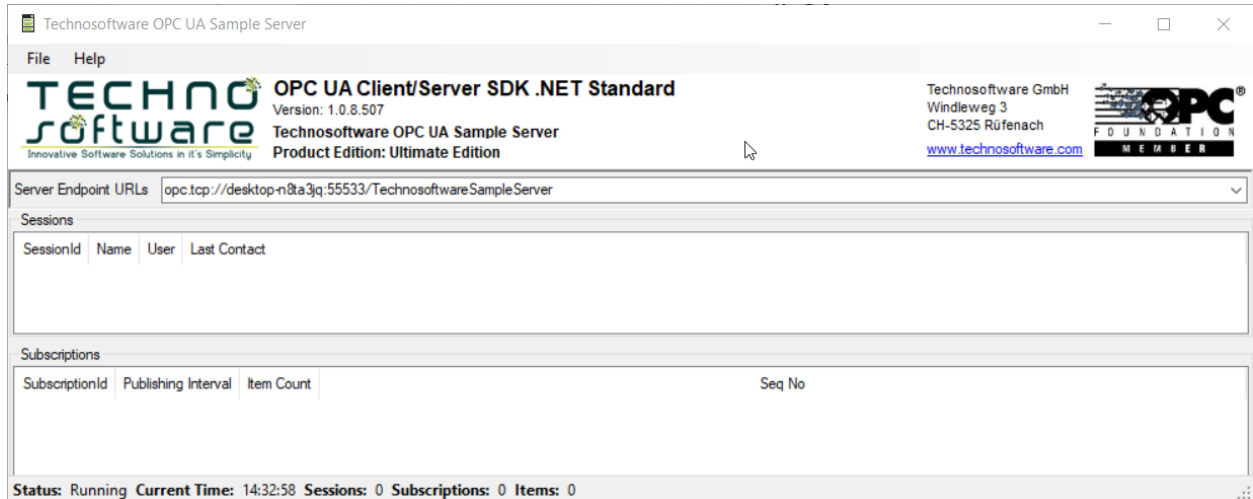

T

3 OPC UA Sample Server and OPC UA Sample Client Usage

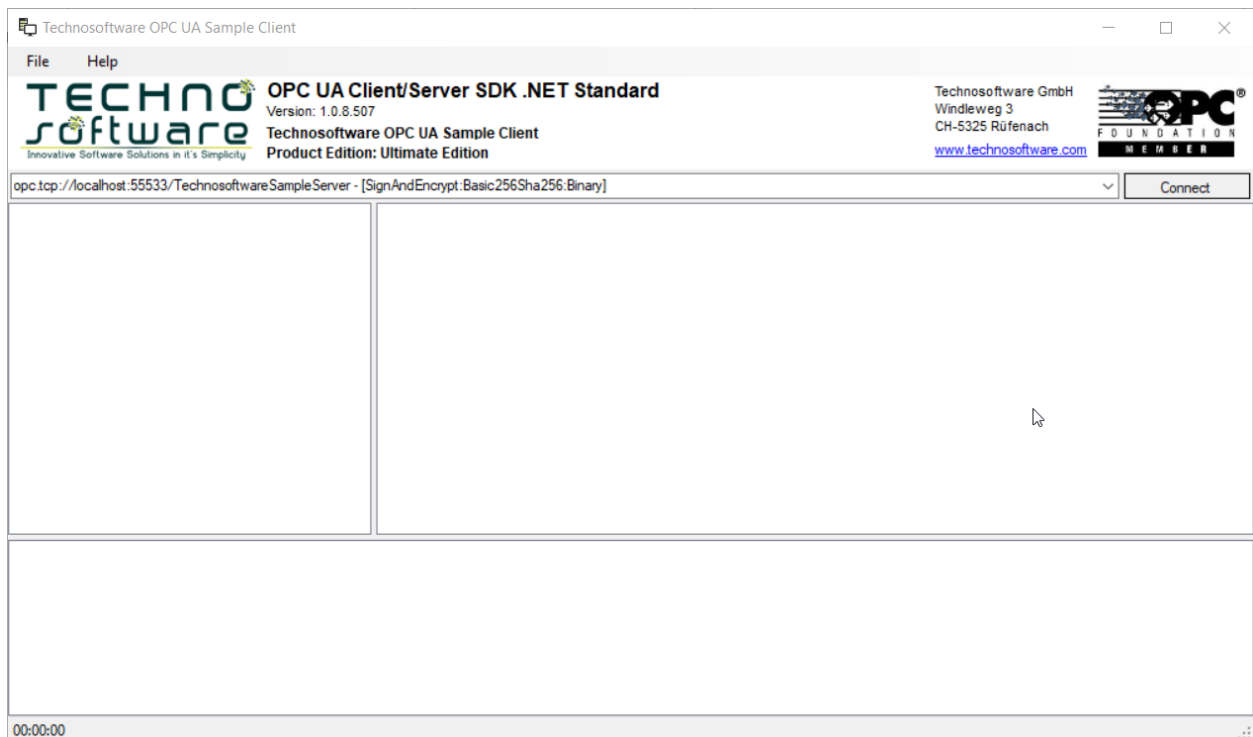
The **OPC UA Bundle SDK .NET Standard** installer contains two OPC UA Applications called:

1. OPC UA Sample Server
2. OPC UA Sample Client

Starting the OPC UA Sample Server should show you the following application:

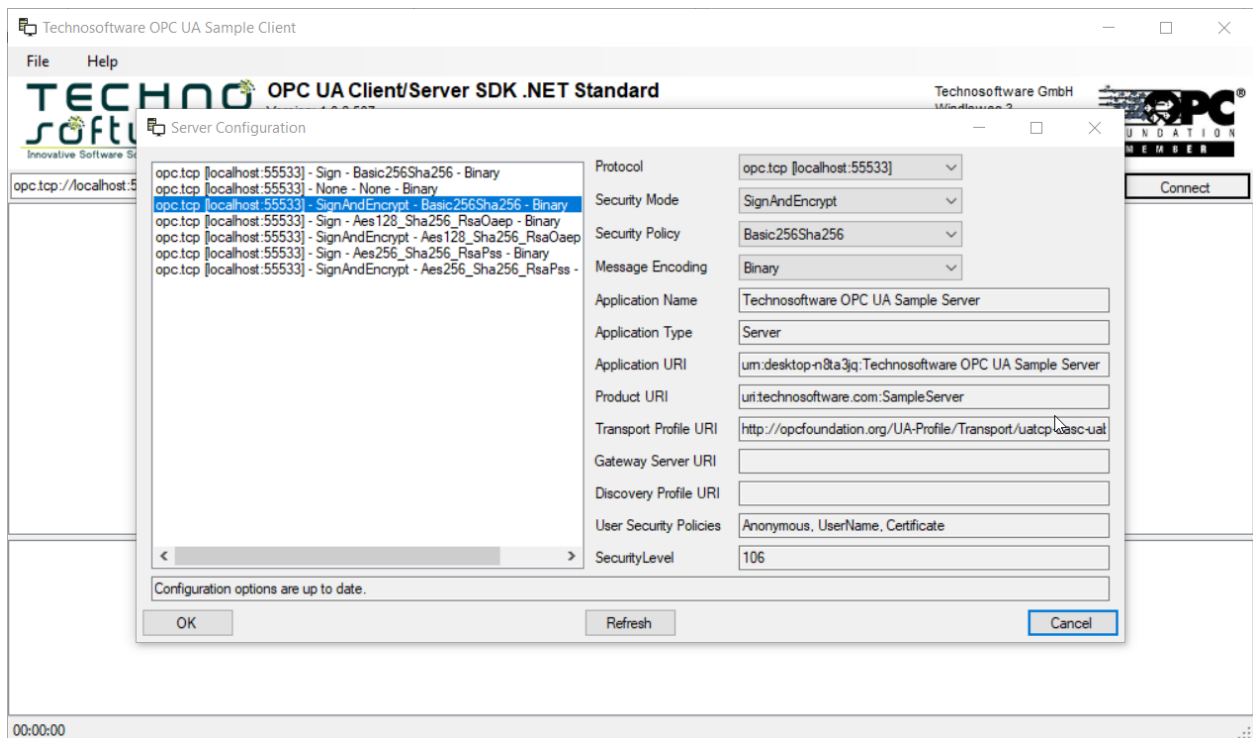


Starting the OPC UA Sample Client should show you the following application:

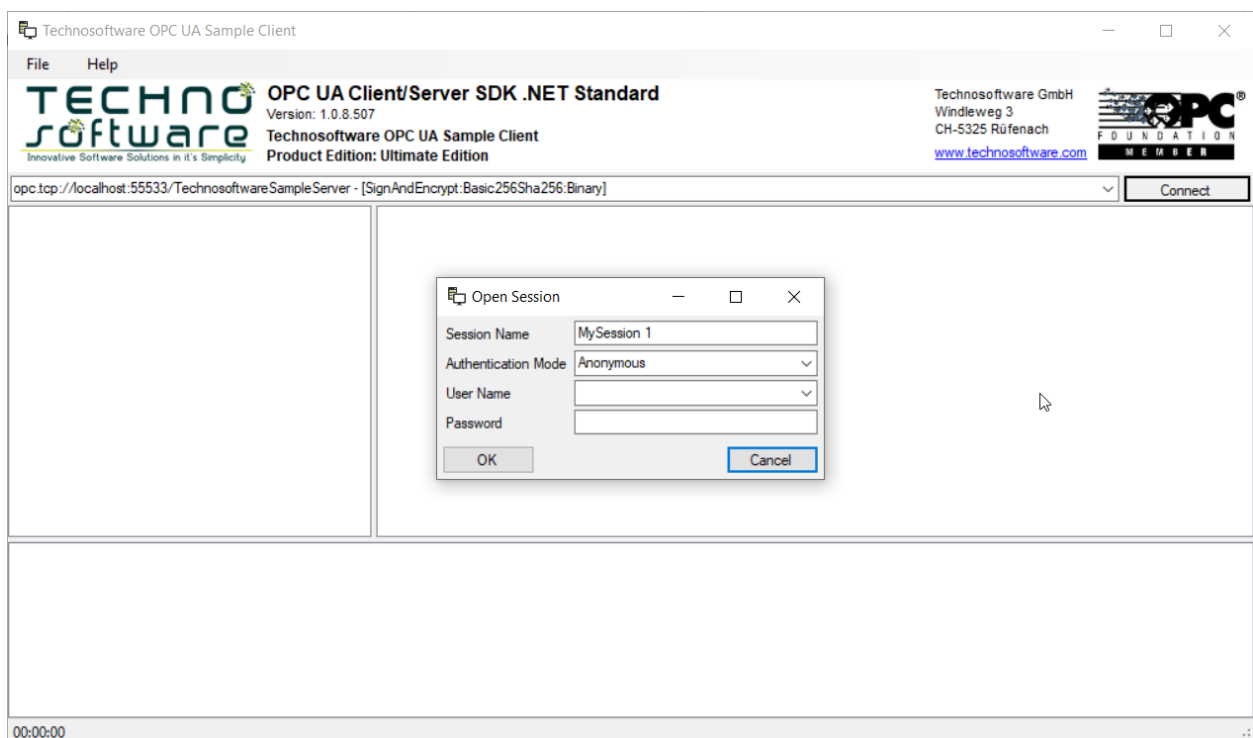


T

The easiest way to connect the client with the server is just pressing the Connect button. The following screen appears:

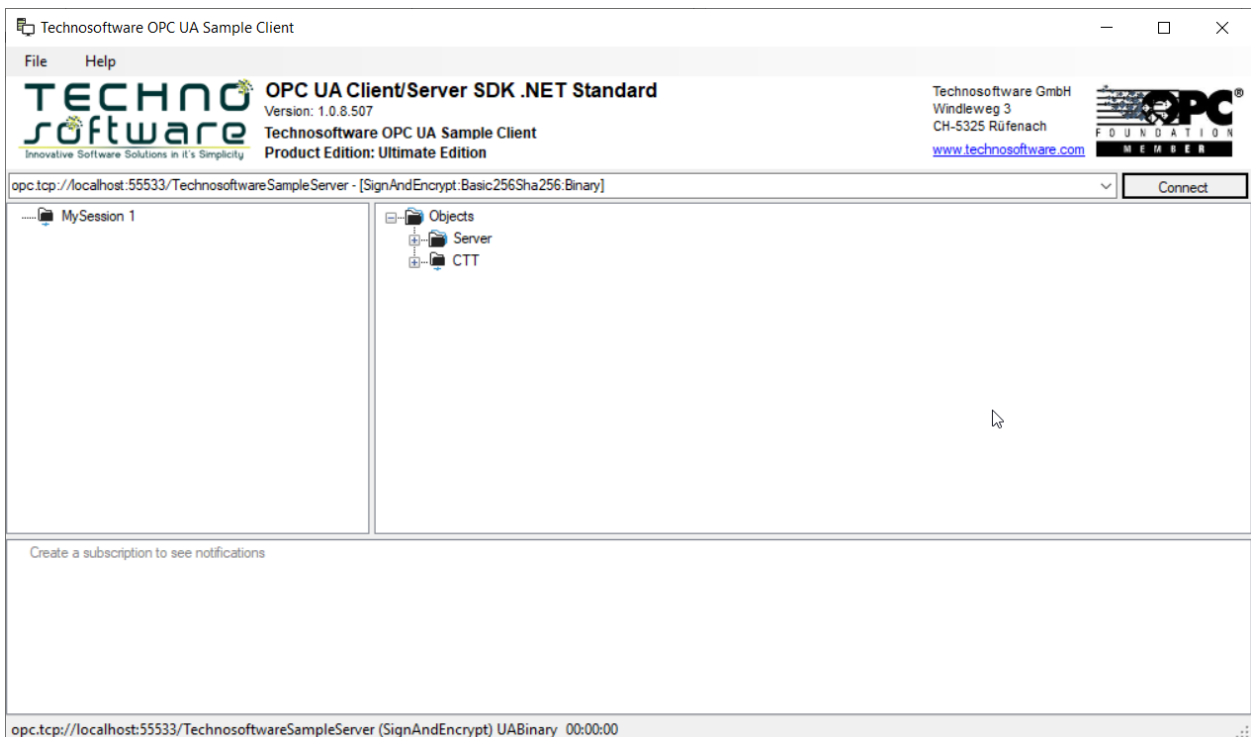


Here you can also just press Ok as first-time user to get to the following screen:

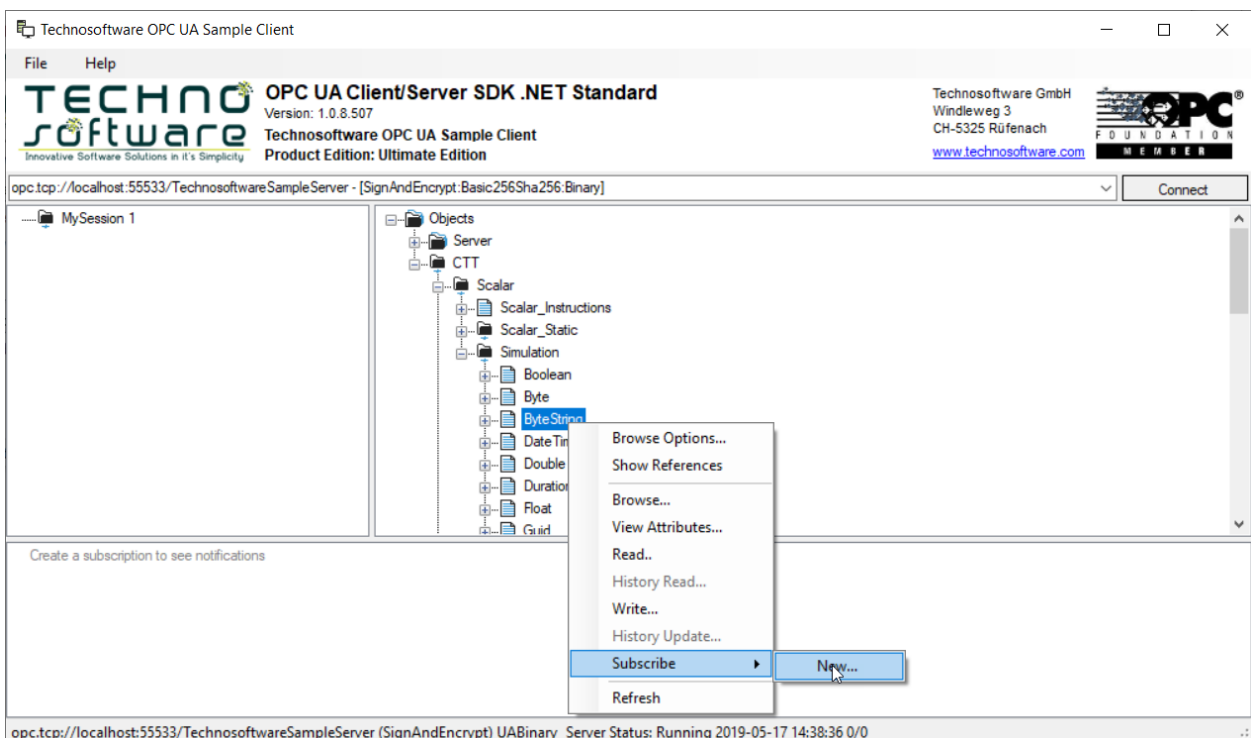


T

And once again, just press Ok:

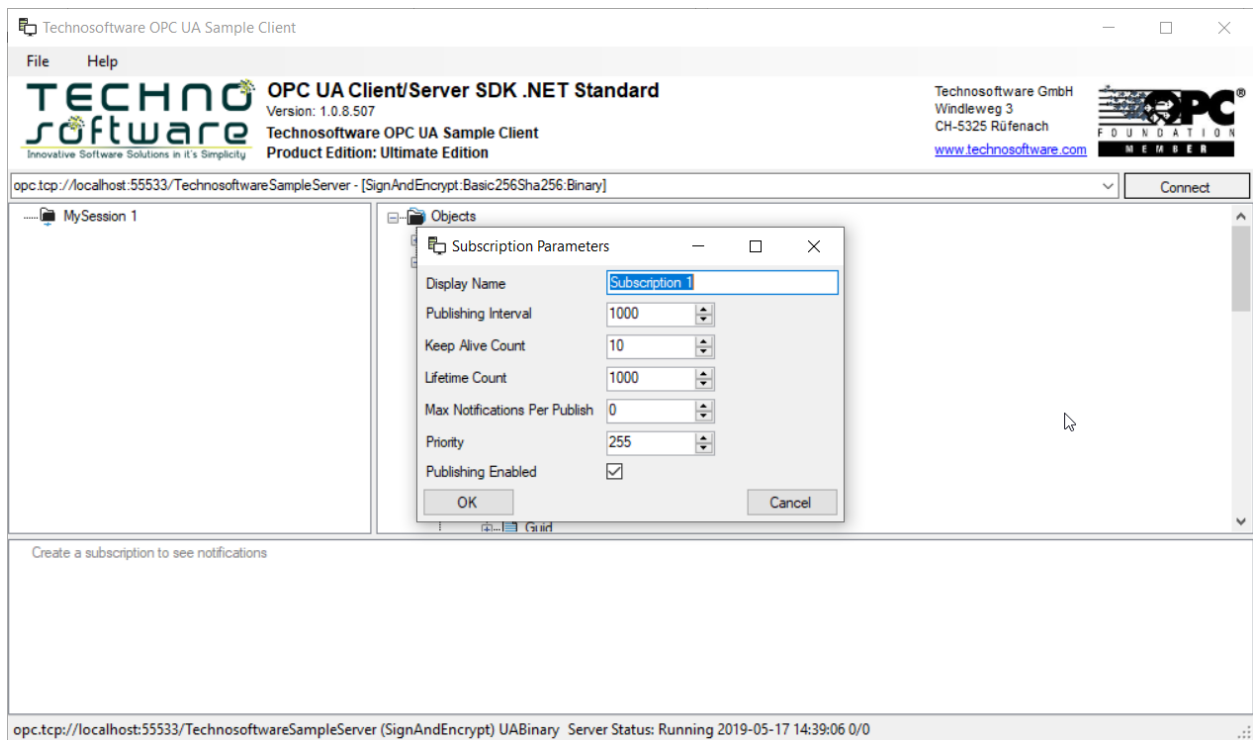


You can now create a subscription to one of the simulated items by browsing to CTT / Scalar / Simulation and then selecting one of the items, e.g. Double as shown in the screen below:

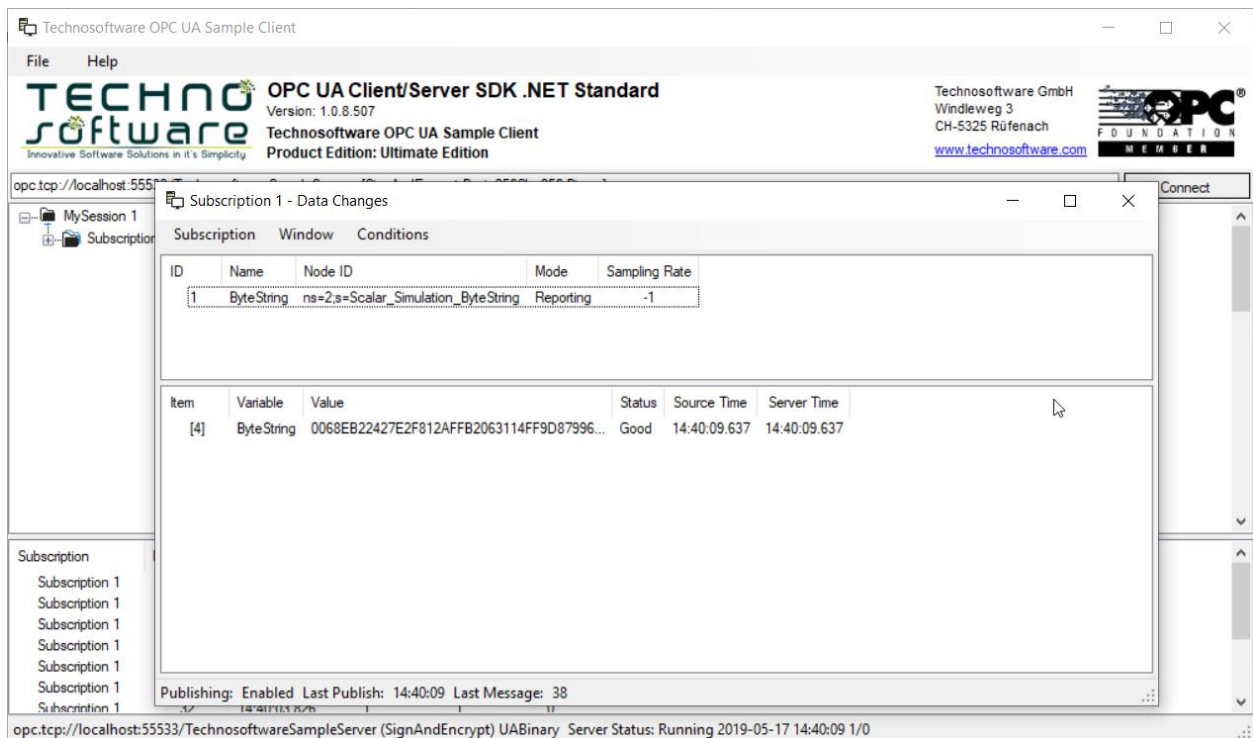


T

Create a subscription with the parameter dialog shown below:



Press Ok and you should get a new dialog with changing values:



4 OPC UA Security

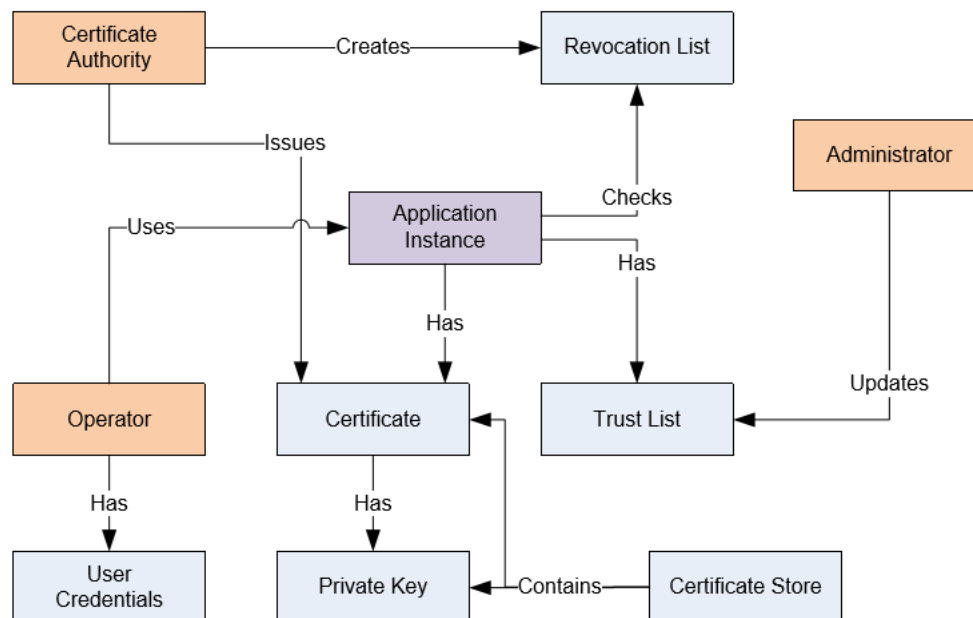
This chapter is based on the Whitepaper ***The OPC UA Security Model for Administrators, Version 1.00 from July, 2010*** ([OPC-UA Security Model for Administrators V1.00.pdf](#)) but contains several adaptations valid for the OPC UA .NET Standard based SDKs and tools.

4.1 Background

A security model is an architecture that allows developers, administrators and end users to use applications in distributed environment while ensuring that the applications, the computers they run on and the information exchanged is not compromised. A complete security model has several facets including application security, transport security, user authorization and authentication and traceability. This white paper describes how to use the OPC UA security model to ensure application and transport security. The target audience for this document are systems administrators and end users. A second whitepaper will discuss the security model from the perspective of a software developer.

The OPC UA security model has been designed to meet the requirements of many different systems while using the same infrastructure. In order to accommodate different security and administrative requirements the OPC UA security model offers four tiers for application authentication and two tiers for certificate management. It is up to the administrator to decide which tiers best match their needs. Applications should support all tiers. This document also discusses the administrative procedures required by a tier. Applications must allow administrators to configure the level of security enforced by their application just like web browsers allow administrators to configure the security level enforced by the browser.

The UA Security Model defines four principal actors: The Application Instance, the Application Administrator, the Application Operator and the Certificate Authority. The relationships between these actors are shown in the following figure. Each of the entities is described in the text that follows.



T

Application Instance - An OPC UA Application installed on a single machine is called an Application Instance. Each instance must have its own Certificate which it uses to identify itself when connecting to other applications. Each Application Instance has a globally unique URI which identifies it.

Administrator - An Application Instance must have an Administrator which manages the security settings for the Application.

Operator - An Operator is person who uses the Application Instance. More than one Operator may exist for any given application. An Operator may have User Credentials which are used to determine access rights and to track activities.

Certificate Authority (CA) - A Certificate Authority (CA) is an administrator or organization which is responsible for creating Certificates. The Certificate Authority must verify that information placed in the Certificate is correct and add a digital signature to the Certificate that is used to verify that the information has not been changed. Each CA must have its own Certificate which is used to create the digital signatures.

Certificate - A Certificate is an electronic ID that can be held by an application. The ID includes information that identifies the holder, the issuer and a unique key that is used to create and verify digital signatures. The syntax of these certificates conforms to the X509 specification, as a result, these certificates are also called "X509 Certificates". Certificates also have a Private Key associated with them.

User Credential - A User Credential is a generic term for an electronic ID which identifies an Operator. It may be passed to a Server after the Application Certificate is used to create a secure channel. It is used to determine access rights and to track activities.

Private Key - A Private Key is a secret number known only to the holder of a Certificate. This secret allows the holder to create digital signatures and decrypt data. If this secret is revealed to unauthorized parties, then the associated Certificate can no longer be trusted or used.

Certificate Store - A Certificate Store is a place where Certificates and Private Keys can be stored on a file system. All Windows systems provide a registry-based store called the Windows Certificate Store. All systems support a directory containing the Certificates stored in a file which is also called an OpenSSL Certificate Store.

Self-Signed Certificate - A Self-Signed Certificate is a Certificate which has no Certificate Authority. These Certificates can be created by anyone and can be used in situations where the administrators of UA Applications are able to verify the claims by reviewing the contents themselves and security is addressed in another manner.

Trust List - A Trust List is a list of Certificates which are trusted by an Application Instance. When security is enabled UA Applications must reject connections from peers if they do not have a Certificate that is in the trusted or issued by a CA that is in the Trust List.

Revocation List - A Revocation List is a list of Certificates which have been revoked by a CA and must not be accepted by an Application Instance.



4.2 Security Tiers

4.2.1 The Basics

In OPC UA, each installation of an application must have an application instance certificate that uniquely identifies the application and the machine that it is running on. These certificates come with private keys that allow applications to create secure communication channels that cannot be viewed by 3rd parties or modified while in transit. These certificates also allow OPC UA applications to be identified by peers and to block communication from a peer if it is not authorized.

4.2.2 Tier 1 - No Authentication

In this tier the client and server allow any peer to communicate which means that all valid certificates are trusted. The application certificates are used only to provide unverifiable information about the peer. The receiver has no way to know if the sender is the legitimate holder of the certificate.

In this mode the client and server automatically accept valid certificates even if they have not been explicitly added to the trust lists managed by the client and server applications. This mode requires no configuration at the server or client.

This tier cannot ensure the privacy of any information transmitted, including user credentials. This tier would only be appropriate in a system that has guaranteed security in some other manner, such as a physically secured and isolated system or where communications is secured via VPN or other such transport layer security. It would also be appropriate in a situation where all information is public and access to it is open.

A developer need only configure an installation procedure that generates an application instance certificate for an application on installation.

4.2.3 Tier 2 - Server Authentication

In this tier the server allows any client to connect and if user authentication is required it is done by sending user credentials such as a username/password after the secure channel has been established. Clients, on the other hand, must be configured by the administrator to trust the Server.

Clients will trust a Server if an administrator has explicitly placed the Server certificate into its trust list or if the Server's certificate was issued by a CA which is in its trust list.

If the Server's certificate was not explicitly in the trust list (i.e. the certificate was issued by a CA that is in the trust list) the client should compare the DNS name in the Server's certificate to the DNS name it used to connect to server. If they match the client knows it is connecting to the machine it thinks it is connecting to. This does not guarantee that the client has connected to correct server, only that the machine is the correct machine.

This tier is used by most Internet banking applications where the bank's web server has a certificate issued by Certificate Authorities like Verisign which are automatically placed in the browsers trust list by the Windows operating system. It provides a fairly good security, but the server cannot restrict the client applications.



4.2.4 Tier 3 - Client Authentication

In this tier the client connects to any server, but the server only allows trusted clients to connect. Clients never provide sensitive information since it does not know if the server is legitimate.

In this tier clients need no pre-configuration other than the URL of the server. However, Servers will only trust clients with certificates that have been placed by administrators in the server's trust list or if the Clients certificate was issued by a CA which is in its trust list

This mode is used by discovery services which need to ensure that only authorized applications have access to them, but clients don't care if the server is not legitimate. The local discovery server (LDS) operates in this mode and only allows authorized applications to register themselves.

4.2.5 Tier 4 - Mutual Authentication

In this tier both the client and server only allow trusted peers to connect. It offers the highest level of security but requires that both the client and server be configured in advance. This is the recommended mode for any public or semi-public deployment of OPC UA or for deployments where security is a primary concern.

As in Tier 2, clients should check the DNS name if the Server certificate was not explicitly placed in the trust list.

It will be used in environments where administrators want complete control over which applications can talk to each other. It also provides the most secure environment.

Application installation should default to Tier 4 mode



4.3 Certificates and Certificate Stores

4.3.1 Overview

When working with certificates it is important to understand the formats associated with a certificate and where the certificates are stored. Both the formats and storage location vary from platform to platform and vary by the cryptographic library that is being used by an application.

4.3.2 Certificates and Private Keys

Certificates are typically stored in files that can have several formats.

The formats used by UA applications are shown in the following tables.

DER	An ASN.1 blob encoded with the DER (distinguished encoding rules). File extension is *.der or *.cer on Windows systems. Use only for storing the Certificate (not the Private Key). Certificates can be imported or exported to/from Windows Certificate Stores using this format. It is also the file format used to store a certificate in a directory store.
PKSC#12	A binary format used to store RSA private keys with their certificates. File extension is *.pfx. May be password protected. Private keys can be imported or exported to/from Windows Certificate Stores using this format.
PEM	A text format used to store private keys File extension is *.pem. May be password protected. This format is only used by some OpenSSL based applications or windows application, but they include items such as CRL lists. Other formats such as *.crt, or *.crl may occur in some systems, but all others can be converted or matched to one of the above, by the operating system

Other formats such as *.crt, or *.crl may occur in some systems, but all others can be converted or matched to one of the above, by the operating system



4.3.3 Windows Certificate Stores

Windows Certificate Stores are accessible via standard Windows tools and WIN32 APIs. They are physically stored in the registry but must be accessed via the standard APIs.

There are two special store locations: LocalMachine and CurrentUser:

- The LocalMachine location contains Certificate stores shared by all users and services on a machine. All users have read access to these stores.
- The CurrentUser (or CurrentService) location contains Certificate stores which are only accessible to the current user or service. These stores can be accessed by administrators via the standard APIs.

The Windows Certificate Stores are identified by the location and a store name separated by a backslash. e.g. LocalMachine\UA Applications or CurrentUser\UA Applications.

Certificates placed in Windows Certificate Stores may have private keys associated with them; however, users will not be able to access these private keys unless they have been granted permission. The UA Certificate Tool can be used to manage permissions for private keys.

Important:

The Windows Certificate Store is no longer supported with the .NET Standard version. Only the original versions of the OPC UA SDKs .NET supported it.



4.3.4 Directory Stores

Any directory can contain *.der, *.pfx or *.pem files. These directories are called a certificate store and identified by the full path of the directory.

By convention the UA Stacks also support a directory store which places the private keys in a separate subdirectory. The subdirectory for certificates is called „certs“ and the subdirectory for private keys is called „private“. This style of certificate store is also called an OpenSSL store.

An OpenSSL store is identified by the full path of the root directory. The presence of the „certs“ subdirectory is used to distinguish between a simple directory store and an OpenSSL store.

OpenSSL may also make use of directories such as `crl`, which would contain certificate revocation lists. This directory is at the same level as the „certs“ directory.

The standard locations of the different used folders are:

ApplicationCertificate:	Where the application instance certificate is stored (MachineDefault): %CommonApplicationData%\OPC Foundation\pki\own
TrustedIssuerCertificates:	Where the issuer certificates are stored (certificate authorities). Typical web browsing applications trust any certificate issued by a CA in the "Trusted Root Certification Authorities" certificate store. However, this approach is not appropriate for UA because Administrators have no control over the CAs that get placed in that Root store to facilitate web browsing. This means Administrators must specify a different store that is used only for UA related CAs and/or they must explicitly specify the certificate for each trusted certification authority: %CommonApplicationData%\OPC Foundation\pki\issuer
TrustedPeerCertificates:	Where the trust list is stored. Some UA applications will use self-signed certificates (certificates without a CA) which means that every application which communicates with it must be configured to trust it. Administrators may designate a certificate store that contains trusted UA application instance certificates (this store should not be the same as the store used for Cas certificates). Alternately, Administrators may enter the certificates explicitly in this list. Note that entries in this list may either reference a certificate in the store or may contained the entire certificate encoded as base64 data: %CommonApplicationData%\OPC Foundation\pki\trusted
RejectedCertificateStore:	The directory used to store invalid certificates for later review by the administrator: %CommonApplicationData%\OPC Foundation\pki\rejected
UserIssuerCertificates:	Where the User issuer certificates are stored: %CommonApplicationData%\OPC Foundation\pki\issuerUser
TrustedUserCertificates:	Where the User trust list is stored: %CommonApplicationData%\OPC Foundation\pki\trustedUser



4.3.5 X509 Stores

With the .NET Standard version, the Windows Certificate Stores was replaced with the Folder & OS-level certificate-store X509Store.

This certificate store is best suited for cross platform support.

4.3.5.1 Windows .Net applications

By default, the self signed certificates are stored in a X509Store called `CurrentUser\UA_MachineDefault`. The certificates can be viewed or deleted with the Windows Certificate Management Console (`certmgr.msc`). The trusted, issuer and rejected stores remain in a folder called `OPC Foundation\CertificateStores` with a root folder which is specified by the SpecialFolder variable `%CommonApplicationData%`. On Windows 7/8/8.1/10 this is usually the invisible folder `C:\ProgramData`.

4.3.5.2 .Net Standard Console applications on Windows, Linux, iOS etc.

The self signed certificates are stored in a folder called **OPC Foundation\CertificateStores\MachineDefault** with a root folder which is specified by the SpecialFolder variable `%LocalApplicationData%` or in a **X509Store** called **CurrentUser\My**, depending on the configuration. For best cross platform support the personal store **CurrentUser\My** was chosen to support all platforms with the same configuration. Some platforms, like macOS, do not support arbitrary certificate stores.

The *trusted*, *issuer* and *rejected* stores remain in a shared folder called **OPC Foundation\CertificateStores** with a root folder specified by the SpecialFolder variable `%LocalApplicationData%`. Depending on the target platform, this folder maps to a hidden location under the user home directory. The standard locations of the different used folders are:

ApplicationCertificate:	Where the application instance certificate is stored (MachineDefault): <code>CurrentUser\My</code>
TrustedIssuerCertificates:	Where the issuer certificates are stored (certificate authorities). Typical web browsing applications trust any certificate issued by a CA in the "Trusted Root Certification Authorities" certificate store. However, this approach is not appropriate for UA because Administrators have no control over the CAs that get placed in that Root store to facilitate web browsing. This means Administrators must specify a different store that is used only for UA related CAs and/or they must explicitly specify the certificate for each trusted certification authority: <code>%LocalApplicationData%/OPC Foundation/pki/issuer</code>
TrustedPeerCertificates:	Where the trust list is stored. Some UA applications will use self-signed certificates (certificates without a CA) which means that every application which communicates with it must be configured to trust it. Administrators may designate a certificate store that contains trusted UA application instance certificates (this store should not be the same as the store used for Cas certificates). Alternately, Administrators may enter the certificates explicitly in this list. Note that entries in this list may either reference a certificate in the store or may contained the entire certificate encoded as base64 data: <code>%LocalApplicationData%/OPC Foundation/pki/trusted</code>
RejectedCertificateStore:	The directory used to store invalid certificates for later review by the administrator: <code>%LocalApplicationData%/OPC Foundation/pki/rejected</code>



4.4 Key Certificate Properties

A certificate contains several fields that have functionality as defined in the x.509 specification. A few key fields are listed here:

Common Name:	This is usually the Application Name obtained from the configuration file associated with the application for which the certificate is being generated, but it must be an appropriate name for the application.
Organization:	This is usually a description of the organization where the application is installed.
SubjectAltName: URI:	This is the unique Application URI associated with the application, and is obtained from the configuration file associated with the application for which the certificate is being generated. Only one URI field may be in a certificate.
SubjectAltName: DNSName IPAddress	The DNS name or IP Address of the machine where this application is installed. Multiple DNS Names and/or IP Addresses can be in a single certificate.
Valid from Valid to:	This is the starting and ending date for which a certificate is valid. The Automatic certificate checking will verify that the current date is between these dates when validating a certificate. The administrator should check these dates periodically and replace any certificates that are about to expire with new certificates.

These fields are usually used by an administrator to help identify and match a certificate to an application. OPC UA applications also automatically check some of these fields when verifying a received certificate from an application. Certificates contain additional information, but this information is usually provided by the tool that generates the certificate.

5 Tools

This section will describe some of the common tools that can be used to manage OPC UA Applications and certificates. The tools include:

- **OPC UA Configuration Tool** – Manages OPC UA Applications, Security Settings, Certificates and more. A general-purpose tool provided by the OPC Foundation that simplifies OPC UA management.
- **UA Certificate Generator** – Generate Application instance certificates. These Certificates can be CA based or self-signed. This tool is written in C++ and uses OpenSSL thus can be adapted to almost any platform.
Important:
 The OPC UA Certificate Generator is no longer needed with the .NET Standard version. Only the original versions of the OPC UA SDKs .NET required it.
- **Windows Certificate Management Console** – The certificates can be viewed or deleted with the certmgr.msc.

5.1 OPC UA Configuration Tool

The OPC UA Configuration Tool is also used for managing the applications and its certificates. Certificates and its importances for OPC UA are explained in the different OPC UA specifications as well as a whitepaper. Please see the References mentioned later in this document.

The Whitepaper “The OPC UA Security Model for Administrators - Whitepaper Version 1.00 July 7, 2010” explains in detail the certificate handling but uses the OPC UA Configuration Tool delivered by the OPC Foundation. For products from Technosoftware GmbH please use the configuration tool provided by Technosoftware GmbH. You will find many of the following explanations also in the whitepaper from the OPC Foundation but changed in those part referencing to the OPC UA Configuration Tool.

5.1.1 Overview

The OPC UA Configuration Tool is GUI application that allows Administrators to do the following:

- 1) Manage Applications
- 2) Manage Security
- 3) Manage Application Certificates
- 4) Manage Certificates
- 5) Options - Manage HTTP Access Rules and SSL Bindings

The OPC UA Configuration Tool is included in the **OPC UA Bundle SDK .NET Standard** installation.



5.1.2 Choosing an Application to Manage

The OPC UA Configuration Tool shows a list of Applications allowing the Administrator to select an Application to manage. This list is populated with a list of applications installed on the machine from a set of configuration files placed in the %Common ApplicationData%\Technosoftware\Applications, e.g. C:\ProgramData\Technosoftware\Applications, directory.

The files are XML files that can be generated by the Applications themselves. Vendors that wish to facilitate configuration of their applications can create these files and place them in the directory where they will be found by the tool.

The XML file has the following layout:

```
<ManagedApplication xmlns="http://opcfoundation.org/UA/SDK/Configuration.xsd"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <DisplayName>Technosoftware.SampleServer</DisplayName>
  <ExecutablePath>C:\Program Files\Technosoftware GmbH\OPC Sample Binaries .NET
4.7.2\bin\Technosoftware.SampleServer.exe</ExecutablePath>
  <ConfigurationPath>C:\Program Files\Technosoftware GmbH\OPC Sample Binaries .NET
4.7.2\bin\Technosoftware.SampleServer.Config.xml</ConfigurationPath>
  <Certificate>
    <StoreType>Directory</StoreType>
    <StorePath>%CommonApplicationData%\OPC Foundation\pki\own</StorePath>
    <SubjectName>CN=Technosoftware OPC UA Sample Server, C=CH, S=Aargau, O=Technosoftware
GmbH, DC=thomasjohana510</SubjectName>
    <Thumbprint>7DFCBE1A185CDCFEDAEEEE5978AFEE1EA7BA05487</Thumbprint>
  </Certificate>
  <TrustList>
    <StoreType>Directory</StoreType>
    <StorePath>%CommonApplicationData%\OPC Foundation\pki\trusted</StorePath>
  </TrustList>
</ManagedApplication>
```

Each of the fields is described in the following table:

DisplayName	A human readable name for the application. This is displayed in the list.
ExecutablePath	The full path to the executable file.
ConfigurationFile	The full path to the configuration file. The tool can modify the file if it is compatible with the SecuredApplication schema defined in Part 6 - Annex D. If the tool cannot parse the file, it can still be used to control access to the file. All applications built with the UA .NET SDK have a compatible configuration schema.
Certificate	The location of the application instance certificate.
StoreType	The type of certificate store. Must be Directory or X509Store
StorePath	The location of the certificate store.

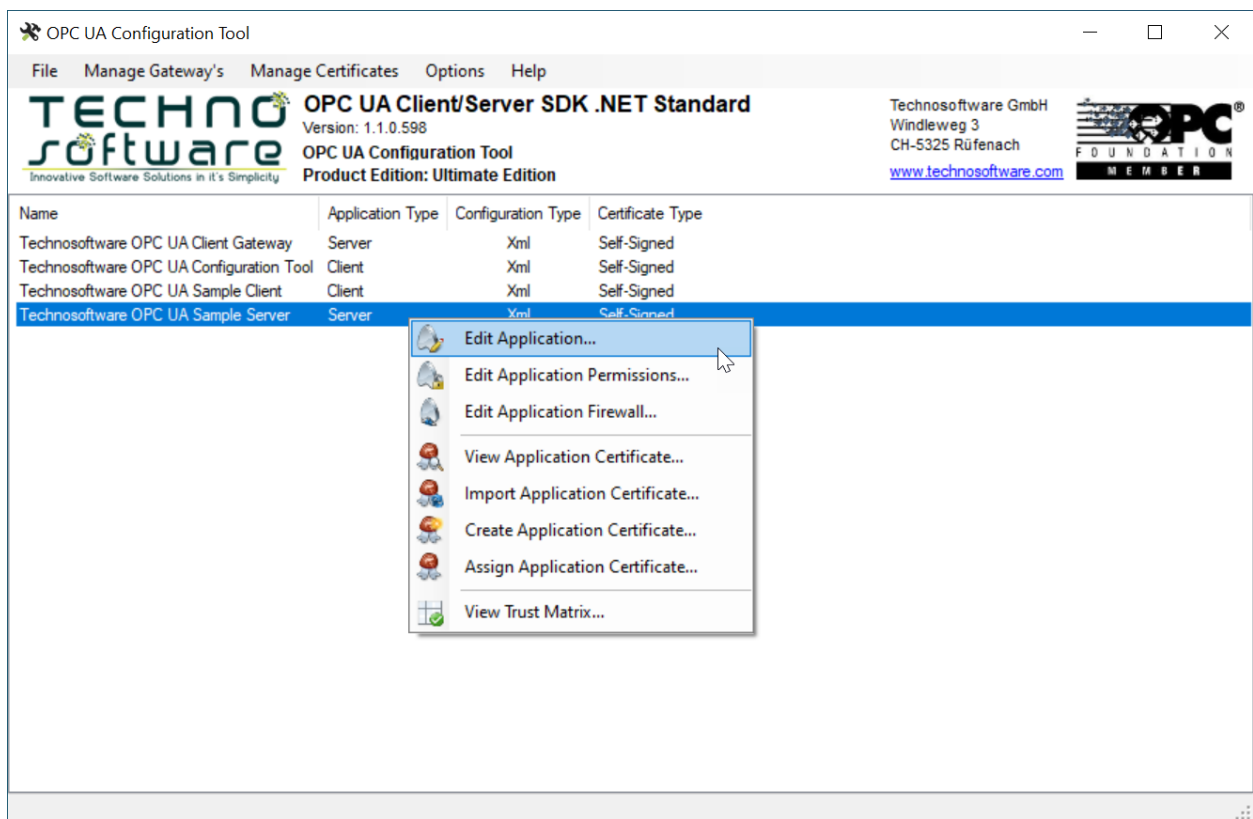
T

SubjectName	The CommonName or full SubjectName in the certificate. The CommonName is the portion of the SubjectName prefixed with „CN=“.
Thumbprint	The SHA1 thumbprint for the certificate.
TrustList	The location of the application trust list.
StoreType	The type of certificate store. Must be Directory or X509Store
StorePath	The location of the certificate store.

5.1.3 Manage Application

By right clicking on an application entry an administrator can review and manage the application settings associated with the selected application instance. Options for this are

- Edit Application
- Edit Application Permissions
- Edit Application Firewall



5.1.3.1 Edit Application

This option displays information about the location of the application settings and allows you to modify them.

Modify Application Information

Application Name:

Executable File:

Configuration File:

Certificate:

Trust List:

This dialog specifies the location of the information needed to configure security for an OPC UA application. The configuration file is the file used by the application to store its security settings. This configuration file can be read by this tool if it conforms to the ApplicationConfiguration schema used by the OPC UA SDK .NET. If it is not known or it uses an unknown schema then the application certificate and trust list must be specified manually. Once this is done the tool can be used to manage the contents of trust list.

5.1.3.2 Edit Application Permissions

This option displays information about the permissions of the application and allows you to modify them.

Manage Application Access Rules

Accounts which are allowed to Change Permissions

Name	Type	Secured Objects
BUILTIN\Administrators	Allow	ExecutableFile, DotNetAppConfigFile, ConfigurationFile, PrivateKey, TrustList
NT AUTHORITY\SYSTEM	Allow	DotNetAppConfigFile, ConfigurationFile, TrustList

Accounts which are allowed to Update Configuration

Name	Type	Secured Objects
BUILTIN\Administrators	Allow	ExecutableFile, DotNetAppConfigFile, ConfigurationFile, PrivateKey, TrustList
NT AUTHORITY\SYSTEM	Allow	DotNetAppConfigFile, ConfigurationFile, TrustList
BUILTIN\Users	Allow	TrustList

Accounts which are allowed to Read Configuration

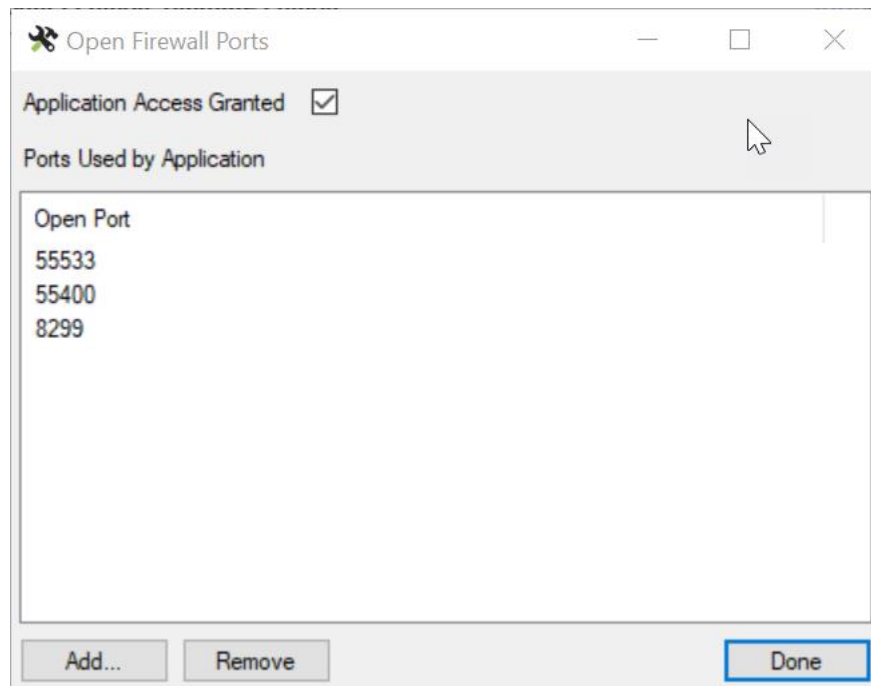
Name	Type	Secured Objects
BUILTIN\Administrators	Allow	ExecutableFile, DotNetAppConfigFile, ConfigurationFile, PrivateKey, TrustList
BUILTIN\Users	Allow	ExecutableFile, DotNetAppConfigFile, ConfigurationFile, TrustList
APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES	Allow	ExecutableFile, DotNetAppConfigFile, ConfigurationFile, TrustList
NT AUTHORITY\SYSTEM	Allow	DotNetAppConfigFile, ConfigurationFile, TrustList
APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES	Allow	DotNetAppConfigFile, ConfigurationFile, TrustList

It is not possible to set permissions on trust lists which are Windows certificate stores. It also may not be possible to permissions on private keys stored in a Windows certificate store on some machines. In these cases, the application should be configured to use directory stores.

T

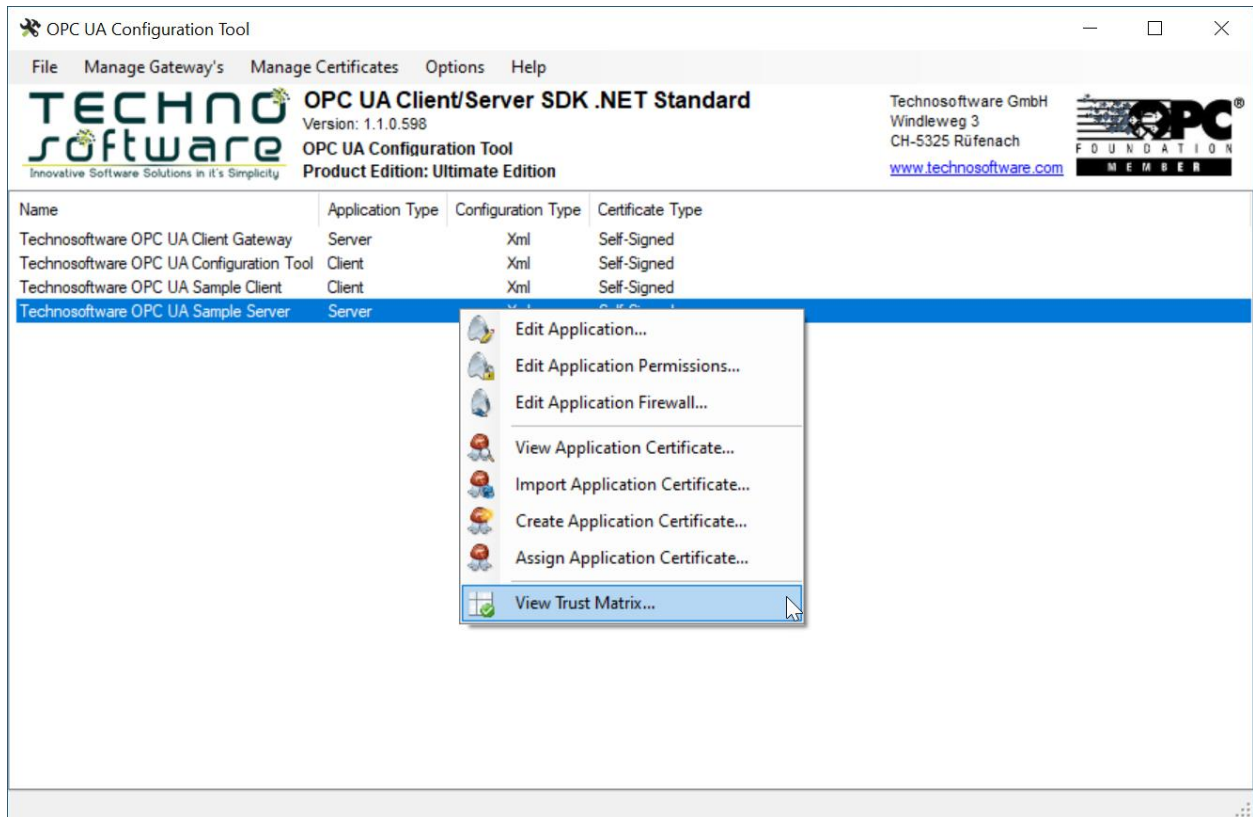
5.1.3.3 Edit Application Firewall

This option displays information about the firewall settings of the application and allows you to modify them.



5.1.4 Manage Security

By right clicking on an application entry an administrator can review and manage the security settings associated with the selected application instance.



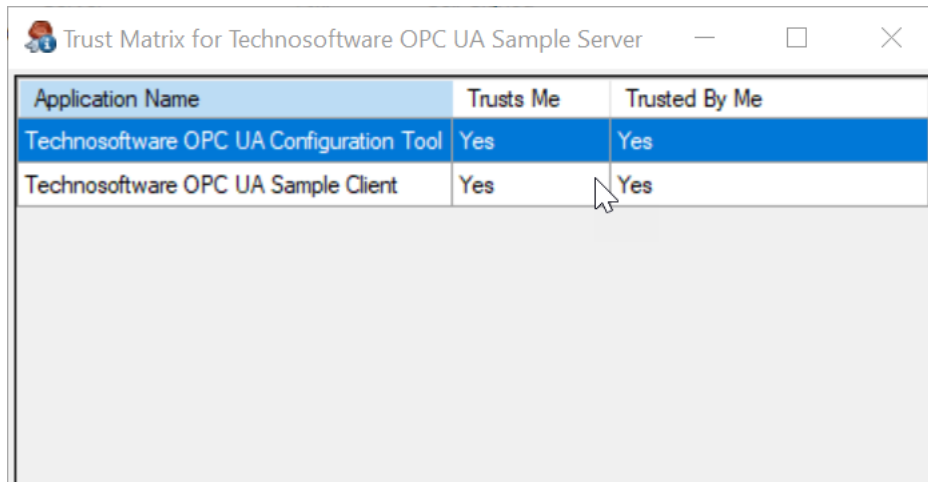
The following configuration options are available for the Application that is selected in the common drop down:

- View Trust Matrix

T

5.1.4.1 View Trust Matrix

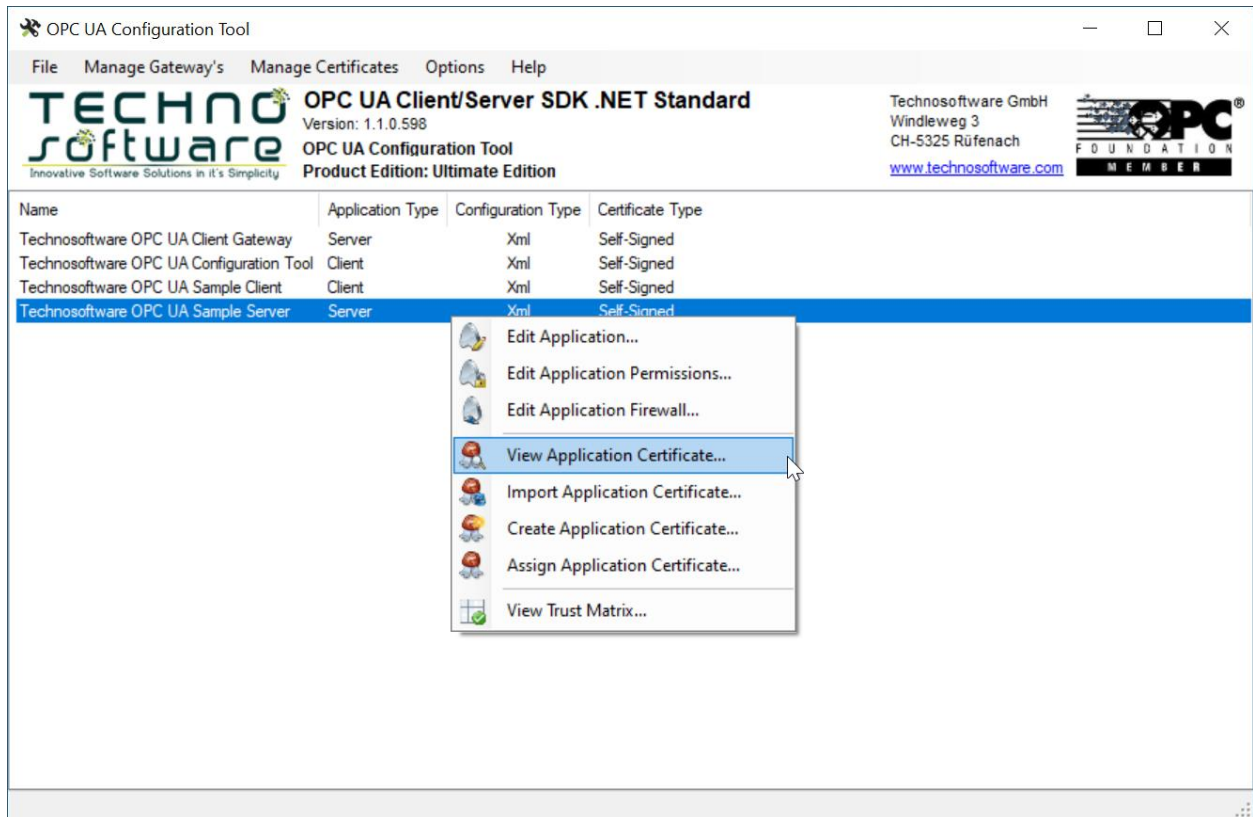
This option displays the list of certificates that the application trusts (see figure below). The administrator can also use the list to ensure that only application that are authorized applications are in the trust list.



Application Name	Trusts Me	Trusted By Me
Technosoftware OPC UA Configuration Tool	Yes	Yes
Technosoftware OPC UA Sample Client	Yes	Yes

5.1.5 Manage Application Certificate

By right clicking on an application entry an administrator can review and manage the application certificates.



The following configuration options are available:

- View Application Certificate
- Import Application Certificate
- Create Application Certificate
- Assign Application Certificate

T

5.1.5.1 View Application Certificate

This option displays the current application certificate in the following dialog:

The screenshot shows a 'View Certificate' dialog box with the following fields and values:

- Store Type:** Directory
- Store Path:** %CommonApplicationData%\OPC Foundation\pki\own
- Application Name:** Technosoftware OPC UA Sample Server
- Organization:** Technosoftware GmbH
- Application URI:** urn.thomasjohana510:Technosoftware OPC UA Sample Server
- Domains:** THOMASJOHANA510
- Subject Name:** CN=Technosoftware OPC UA Sample Server/C=CH/S=Aargau/O=Technosoftware GmbH/DC=thomasjohana510
- Issuer Name:** CN=Technosoftware OPC UA Sample Server/C=CH/S=Aargau/O=Technosoftware GmbH/DC=thomasjohana510
- Valid From:** 2019-06-09 06:39:04
- Valid To:** 2020-06-09 06:39:04
- Thumbprint:** 7DFCBE1A185CDCFEDAEEEE5978AFEE1EA7BA05487

Buttons at the bottom: Details..., Export..., OK, Cancel.

The Export option saves the Certificate to a .DER file. This save does not include the Private Key.

Field Descriptions:

Store Type	The type of certificate repository: Directory or x509Store.
Store Path	The location of the directory store where the certificate is placed.
Application Name	Name of the currently selected application.
Organization	The organization.
Application Uri	A globally unique URI for the application.
Domains	Any specific domain(s) that apply to this application (use ',' separator). Can also use IP address format
Subject Name	The subject name of the certificate. This is a sequence of X500 name-value pairs. e.g. CN=MyApplication,O=MyCompany Each pair is separated by a comma. Values cannot contain commas. The Common Name (CN) should be the applicationName. If not provided, a default value is constructed from the machine name, the applicationName and the organization name. e.g. CN=applicationName,DC=machineName and O=organizationName
Issuer Name	The issuer of the certificate.
Valid From	Start date and time of the certificate.
Valid To	Expiration date and time of the certificate.
Thumbprint	The SHA1 thumbprint for the certificate.

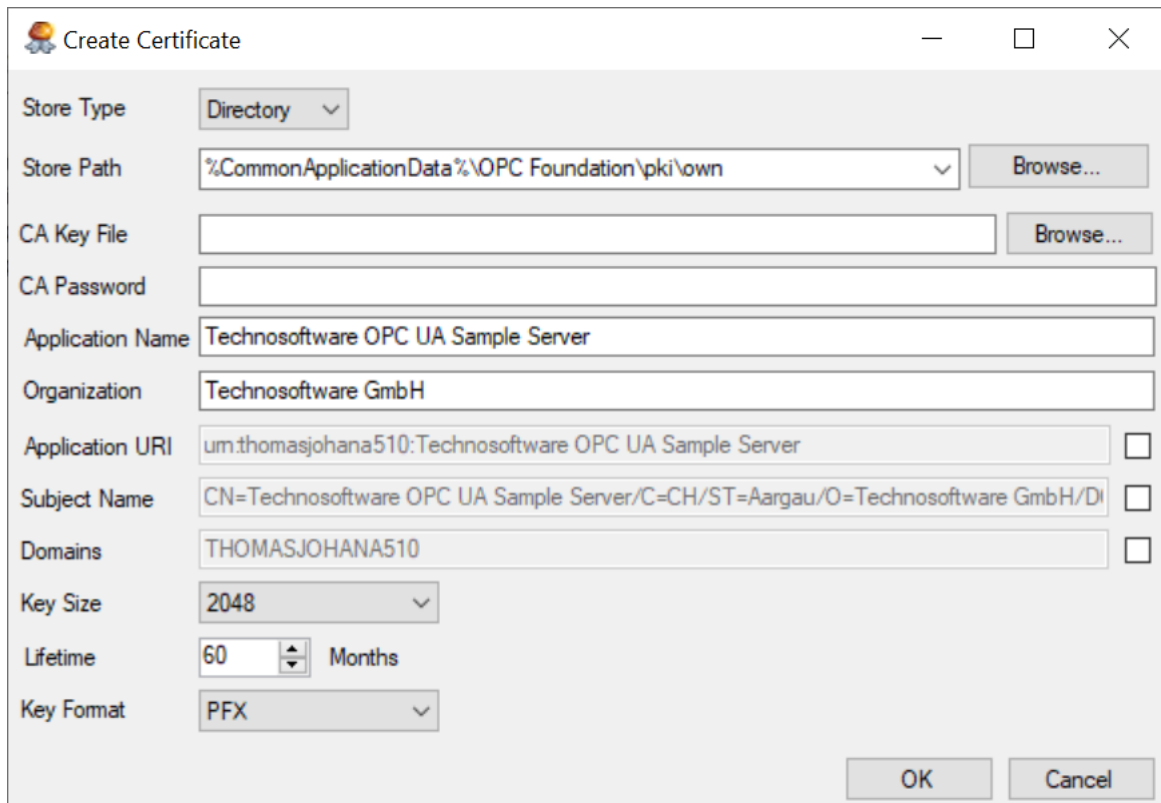
T

5.1.5.2 Import Application Certificate

This option imports an application certificate from a .PFX file stored on disk. It prompts the user to enter a password if one is required. The imported Certificate will replace any Application Instance certificate that may already be assigned to the application.

5.1.5.3 Create Application Certificate

This option creates a new application certificate with the following dialog:



The image shows a 'Create Certificate' dialog box with the following fields and options:

- Store Type:** Directory (dropdown)
- Store Path:** %CommonApplicationData%\OPC Foundation\pki\own (dropdown) with a **Browse...** button.
- CA Key File:** (text field) with a **Browse...** button.
- CA Password:** (text field)
- Application Name:** Technosoftware OPC UA Sample Server
- Organization:** Technosoftware GmbH
- Application URI:** um.thomasjohana510:Technosoftware OPC UA Sample Server (checkbox)
- Subject Name:** CN=Technosoftware OPC UA Sample Server/C=CH/ST=Aargau/O=Technosoftware GmbH/DI (checkbox)
- Domains:** THOMASJOHANA510 (checkbox)
- Key Size:** 2048 (dropdown)
- Lifetime:** 60 (spin box) Months
- Key Format:** PFX (dropdown)
- Buttons:** OK, Cancel

T

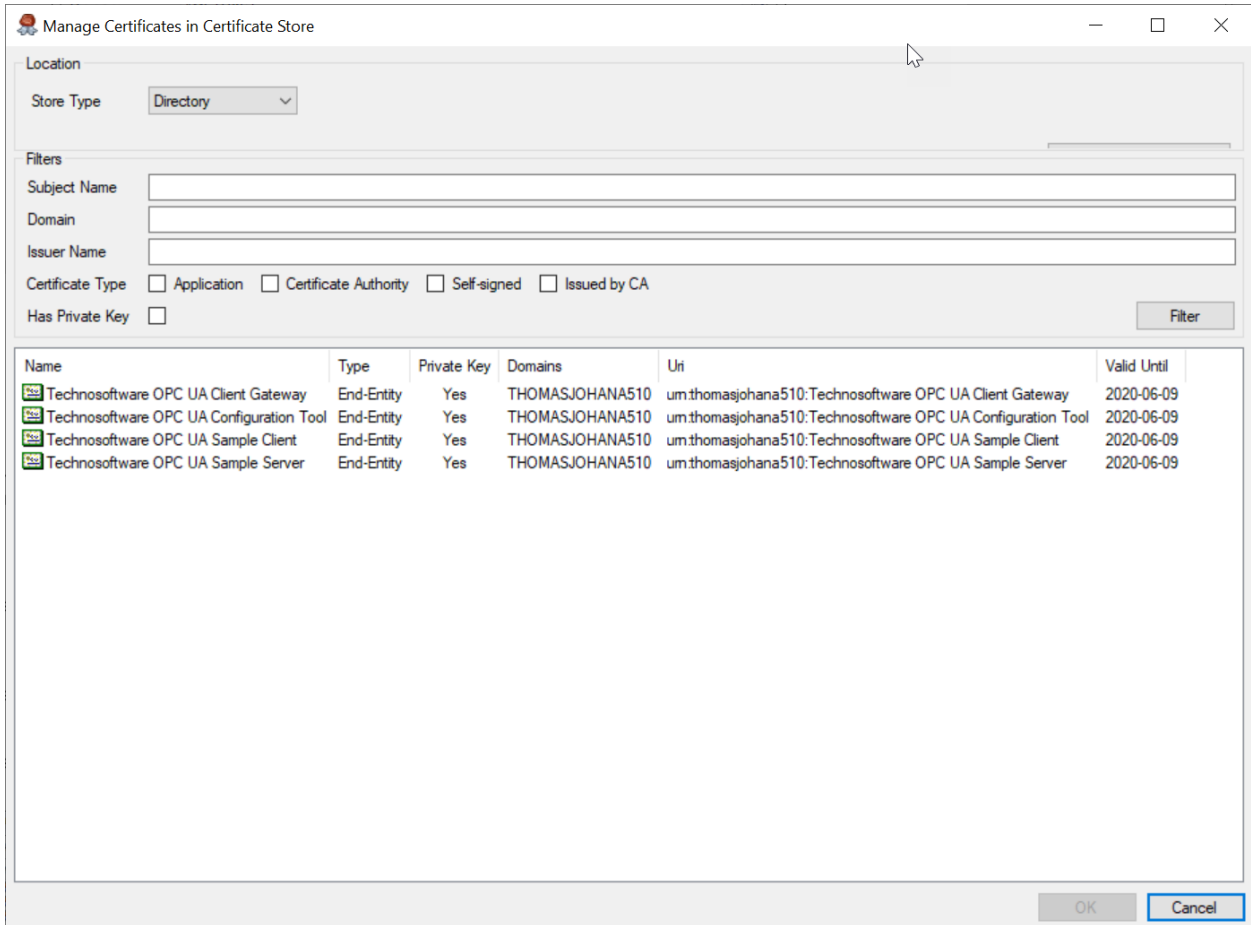
Field Descriptions:

Store Type	The type of certificate repository: Directory or x509Store.
Store Path	The Store Path specifies where the application certificate will be place after it is created.
CA Key File	The CA Key File is a PFX file containing the Certificate Authority private key. If left blank a self-signed certificate is created. If provided the CA Password may also be required. If the Application URI/Subject Name/Domains checkboxes are unchecked the tool will generate them automatically from the other information provided.
CA Password	The password required to access the "CA Key File".
Application Name	Name of the currently selected application.
Organization	The organization.
Application Uri	A globally unique URI for the application.
Subject Name	The subject name of the certificate. This is a sequence of X500 name-value pairs. e.g. CN=MyApplication,O=MyCompany Each pair is separated by a comma. Values cannot contain commas. The Common Name (CN) should be the applicationName. If not provided, a default value is constructed from the machine name, the applicationName and the organization name. e.g. CN=applicationName,DC=machineName and O=organizationName
Domains	Any specific domain(s) that apply to this application (use ',' separator). Can also use IP aaddress format
Key Size	Size of the encryption key. The larger the size, the more complex the algorithm and overhead consumed.
Lifetime	Expiration of the certificate, in months.
Key Format	The format of the certificate file. Use PEM for maximum portability.

T

5.1.5.4 Assign Application Certificate

This option assigns an existing certificate that is already within the certificate repository. The following dialog is shown:



The dialog box 'Manage Certificates in Certificate Store' is shown. It has a 'Location' section with 'Store Type' set to 'Directory'. Below this is a 'Filters' section with input fields for 'Subject Name', 'Domain', and 'Issuer Name'. There are also checkboxes for 'Certificate Type' (Application, Certificate Authority, Self-signed, Issued by CA) and 'Has Private Key'. A 'Filter' button is located to the right of the filters. The main area is a table listing certificates. The table has columns: Name, Type, Private Key, Domains, Uri, and Valid Until. There are four certificates listed, all from 'Technosoftware OPC UA' and valid until '2020-06-09'. At the bottom right are 'OK' and 'Cancel' buttons.

Name	Type	Private Key	Domains	Uri	Valid Until
Technosoftware OPC UA Client Gateway	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoftware OPC UA Client Gateway	2020-06-09
Technosoftware OPC UA Configuration Tool	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoftware OPC UA Configuration Tool	2020-06-09
Technosoftware OPC UA Sample Client	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoftware OPC UA Sample Client	2020-06-09
Technosoftware OPC UA Sample Server	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoftware OPC UA Sample Server	2020-06-09

Simply select a certificate from the list and then click OK.

5.1.6 Manage Certificates

This menu allows an administrator to review and manage the certificates on a machine. This includes the ability to act as a Certificate Authority (CA). The administrator must provide the key file for a CA certificate, it acting as a CA.

This menu provides access to the following options:

- Local Certificate Stores
- Create Certificate Authority

5.1.6.1 Local Certificate Stores

This option views the certificates in the currently selected certificate store. It allows an administrator to view, filter, manage and even delete certificates in multiple certificates stores.

Manage Certificates in Certificate Store

Location

Store Type: Directory

Store Path: %CommonApplicationData%\OPC Foundation\pki\own

Filters

Subject Name:

Domain:

Issuer Name:

Certificate Type: ☐ Application ☐ Certificate Authority ☐ Self-signed ☐ Issued by CA

Has Private Key: ☐

Filter

Name	Type	Private Key	Domains	Uri	Valid Until
Technosoft OPC UA Client Gateway	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoft OPC UA Client Gateway	2020-06-09
Technosoft OPC UA Configuration Tool	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoft OPC UA Configuration Tool	2020-06-09
Technosoft OPC UA Sample Client	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoft OPC UA Sample Client	2020-06-09
Technosoft OPC UA Sample Server	End-Entity	Yes	THOMASJOHANA510	um.thomasjohana510:Technosoft OPC UA Sample Server	2020-06-09

OK Cancel

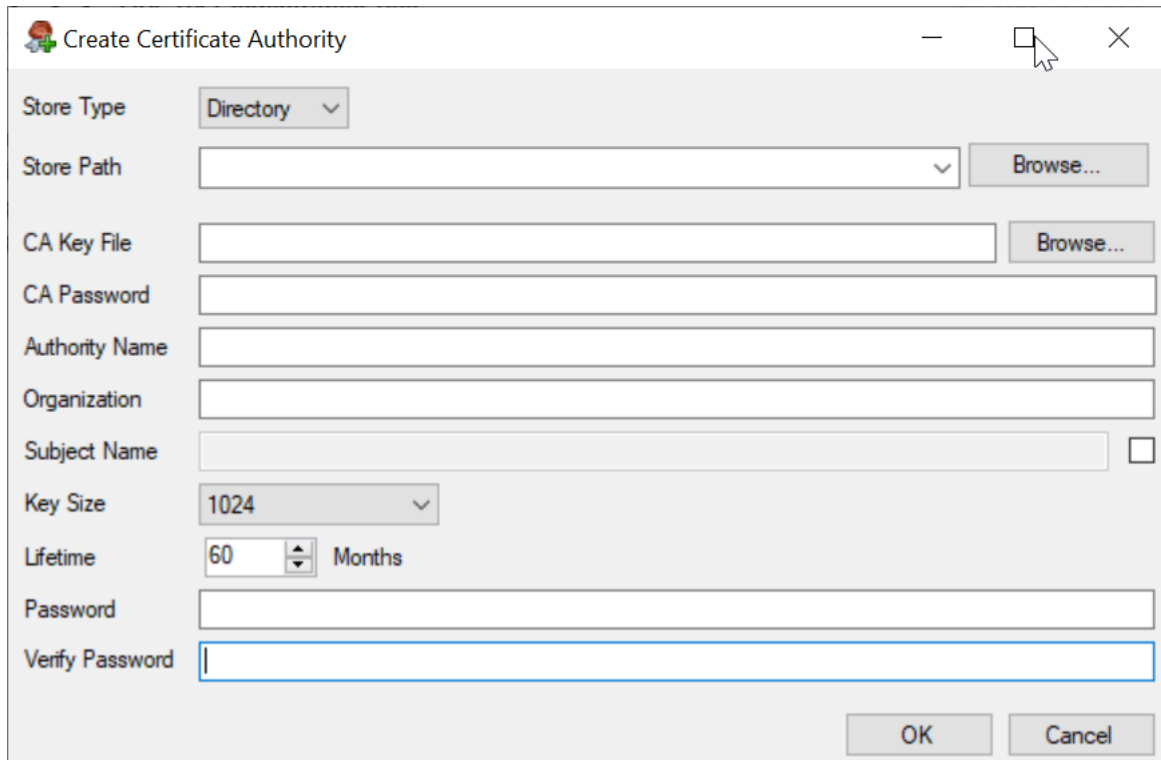
T

5.1.6.2 Create a Certificate Authority Certificate

This menu creates a certificate authority.

Note that the private key for a CA certificate must be carefully protected. It is strongly recommended that a password or a directory with restricted access is used.

The CA certificate can be a sub-CA or a standalone CA. This allows the administrator to create a tree of Certificates Authorities, which may be useful in a large industrial setting where each portion of the plant may have multiple machines and UA applications and trusts need to be established between all of the machines, but the various portions of the plant should be kept separate, with only a select few administrative UA Applications with access to all aspects of the plant.



The screenshot shows a Windows-style dialog box titled "Create Certificate Authority". It contains the following fields and controls:

- Store Type:** A dropdown menu currently set to "Directory".
- Store Path:** A text input field with a "Browse..." button to its right.
- CA Key File:** A text input field with a "Browse..." button to its right.
- CA Password:** A text input field.
- Authority Name:** A text input field.
- Organization:** A text input field.
- Subject Name:** A text input field with a small square checkbox to its right.
- Key Size:** A dropdown menu currently set to "1024".
- Lifetime:** A numeric input field set to "60" with a "Months" label to its right.
- Password:** A text input field.
- Verify Password:** A text input field.
- Buttons:** "OK" and "Cancel" buttons at the bottom right.

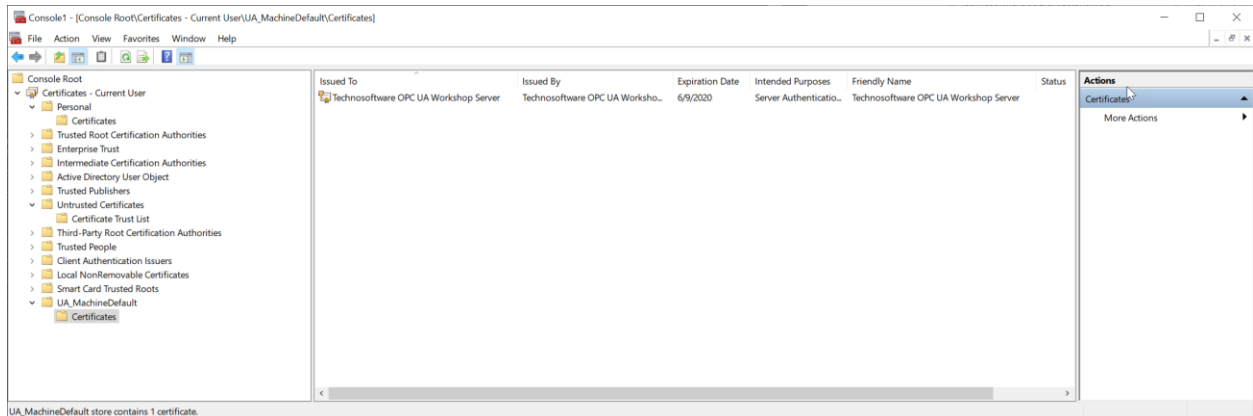
T

5.2 Microsoft Management Console

The Microsoft Management Console is a GUI application that allows Administrators to do the following:

- Update Application Trust Lists
- Manage Certificate Permissions

To start-up certificate management via the Microsoft Management Console (MMC), the user must first configure the add-in for certificate management in the MMC. This can vary between versions of the operating system, see Microsoft for details. For detailed instructions on using this tool please refer to the appropriate Microsoft documentation.



T

Why Technosoftware GmbH?...

Professionalism

Technosoftware GmbH is, measured by the number of employees, truly not a big company. However when it comes to flexibility, service quality, and adherence to schedules and reliability, we are surely a great company which can compete against the so called leaders in the industry. And this is THE crucial point for our customers.

Continuous progress

Lifelong learning and continuing education is, especially in the information technology, essential for future success. Concerning our customers, we will constantly accepting new challenges and exceeding their requirements again and again. We will continue to do everything to fulfill the needs of our customers and to meet our own standards.

High Quality of Work

We reach this by a small, competent and dynamic team of coworkers, which apart from the satisfaction of the customer; take care of a high quality of work. We concern the steps necessary for it together with consideration of the customers' requirements.

Support

We support you in all phases – consultation, direction of the project, analysis, architecture & design, implementation, test and maintenance. You decide on the integration of our coworkers in your project; for an entire project or for selected phases.

Technosoftware GmbH

Windleweg 3, CH-5235 Rüfenach

sales@technosoftware.com

www.technosoftware.com

