OPC UA Solution .NET Installation

Installation and Administration of .NET based OPC UA Applications

T

Document Control

Version	Date	Comment
1.0.8	18-MAY-2019	Initial version based on V1.0.8
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	- 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)
1.3.0	05-JAN-2020	- Added informationen for .NET Core 2.1 - Added information for .NET 4.8
1.4.0	24-JUN-2020	Updated to V 1.4
1.4.10	16-OCT-2020	Updated with information for .NET Core 3.1
2.0	14-NOV-2020	Updated to V2.0
2.1	06-JAN-2021	Added support for .NET 5.0
2.2	22-MAR-2021	Updated to V2.2
2.3	23-APR-2022	Updated to V2.3
2.4	24-AUG-2022	Updated to V2.4

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 .NET or the OPC UA Server .NET. The target audience for this document are systems administrators.

T

Referenced OPC Documents

Documents
Online versions of OPC UA specifications and information models. The OPC UA Online Reference is available at: https://reference.opcfoundation.org
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

T

TABLE OF CONTENTS

T

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

The OPC UA Client & Server .NET can be used not only on Windows but also on Linux and macOS. The Solutions includes a client and server application targeting .NET Core 3.1 or .NET 6.0. 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 Core 3.1 or .NET 6.0 is required. The article describes the installation of .NET 6.0.400 for Windows, Linux and macOS. This version also works with the OPC UA Client and Server Solutions we provide in the GitHub repository at https://github.com/technosoftware-gmbh/opcua-solution-net-samples

Please follow at least the sections.

- Intro
- Download and Install

to install the .NET SDK. You find the different .NET versions also at https://dotnet.microsoft.com/en-us/download/dotnet

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

- OPC UA Client Development with .NET
- OPC UA Server Development with .NET

2 Installation OPC UA Solution .NET

For starting with OPC UA Development you can download the **OPC UA Solution .NET** from:

OPC UA Solution .NET
The OPC UA Solution .NET offers a fast and easy access to the OPC UA Client & Server technology. Develop OPC compliant UA Clients and Servers with C# targeting .NET 6.0 or .NET Core 3.1. For backward compatibility we also provide .NET 4.8, .NET 4.7.2 and .NET 4.6.2 support.
You can download it from https://github.com/technosoftware-gmbh/opcua-solution-net-samples

This GitHub repository is automatically tested with the following environments:

   a. Linux Ubuntu 16.04
      - .NET 6.0.400
      - Mono 6.12
      - MS Build 17.3.0
   b. Mac OS X 10.13
      - .NET 6.0.400
      - Mono 6.12
      - XCode 13
   c. Windows Server 2019
      - .NET 6.0.400
      - Visual Studio 2022 17.2

2.1 Directory Structure

The repository contains the following basic directory layout:

- **bin/**
   - **modelcompiler/**
     Executable of OPC Foundation Model compiler
- **documentation/**
  Additional documentation like:
   - **OPC_UA_Solution_NET_Installation_Guide.pdf**
     Installation of development and run-time system
   - **OPC_UA_Solution_NET_Introduction.pdf**
     Introduction in Developing OPC UA Clients and OPC UA Servers with C# / VB.NET
   - **OPC_UA_Client_Development_with_NET.pdf**
     Tutorial for Developing OPC UA Clients with C# for of .NET 6.0 and .NET Core 3.1
   - **OPC_UA_Server_Development_with_NET.pdf**
     Tutorial for Developing OPC UA Servers with C# for of .NET 6.0 and .NET Core 3.1
- **examples/**
  Sample applications
- **schema/**
  XSD files like the UAModelDesign.xsd used for the Model Designer.
- **Workshop/**
  OPC UA Workshop content as PDF

2.2 DLLs used by applications.

The solution consists of the following main components.

- **Technosoftware.UaCore.dll**
- **Technosoftware.UaBindings.Https.dll**
- **Technosoftware.UaConfiguration.dll**

These two DLLs are used by all applications using the solution. 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.

These DLLs are delivered via NuGet Packages:

- Technosoftware.UaSolution.UaClient
  The OPC UA Client .NET offers a fast and easy access to the OPC UA Client technology. Develop OPC UA 1.00, 1.01, 1.02, 1.03 and 1.04 compliant UA Clients with C#.
- Technosoftware.UaSolution.UaServer
  The OPC UA Server .NET offers a fast and easy access to the OPC Unified Architecture (UA) technology. Develop OPC UA 1.00, 1.01, 1.02, 1.03 and 1.04 compliant Servers with C#.

2.3 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. You can download it via

https://opcfoundation.org/developer-tools/samples-and-tools-unified-architecture/local-discovery-server-lds/

3 Test your installation with .NET Core

The main OPC UA Solution can be found in the root of the repository and is named.

- TutorialSamples.sln

The solution contains several sample clients, as well as several sample server examples used by these clients.

3.1 Prerequisites

Once the *dotnet* command is available, navigate to the following folder:

```
/
```

and execute

```
dotnet restore TutorialSamples.sln
```

This command restores the tree of dependencies.

3.2 Start the server

1. Open a command prompt.

2. Navigate to the folder examples/Simulation/SampleServer.

3. To run the server sample type

   dotnet run --no-restore --framework netcoreapp3.1 --project SampleCompany.SampleServer.csproj -a

   - The server is now running and waiting for connections.

   - The -a flag allows to auto accept unknown certificates and should only be used to simplify testing.

3.3 Start the client

1. Open a command prompt.

2. Navigate to the folder examples/Simulation/SampleClient.

3. To run the client sample type

   dotnet run --no-restore --framework netcoreapp3.1 --project SampleCompany.SampleClient.csproj -a

   - The client connects to the OPC UA console sample server running on the same host.

   - The -a flag allows to auto accept unknown certificates and should only be used to simplify testing.

4. If not using the -a auto accept option, on first connection, or after certificates were renewed, the server may have refused the client certificate. Check the server and client folder %LocalApplicationData%/OPC Foundation/pki/rejected for rejected certificates. To approve a certificate copy it to the %LocalApplicationData%/OPC Foundation/pki/trusted.

3.4 Check the output

If everything was done correctly the client should show the following lines:

```
SampleCompany .NET Core OPC UA Sample Client

Connecting...

Browse address space.

Reading nodes...

    Read Value = {24.08.2022 07:40:44 | 24.08.2022 07:49:04 | Running | Opc.Ua.BuildInfo | 0 | } ,
StatusCode = Good

    Read Value = StartTime , StatusCode = Good

    Read Value = 24.08.2022 07:40:44 , StatusCode = Good

Read a single value from node ns=2;s=Scalar_Simulation_Number.

    Node ns=2;s=Scalar_Simulation_Number Value = 37848 StatusCode = Good.

Read multiple values from different nodes.

    Node ns=2;s=Scalar_Simulation_Number Value = 37848 StatusCode = Good.

    Node ns=2;s=Scalar_Static_Integer Value = 2142943445 StatusCode = Good.

    Node ns=2;s=Scalar_Static_Double Value = 1033833323429888 StatusCode = Good.

Read multiple values asynchronous.

Running...Press Ctrl-C to exit...

Status of Read of Node ns=2;s=Scalar_Simulation_Number is: 37848

Status of Read of Node ns=2;s=Scalar_Static_Integer is: 2142943445

Status of Read of Node ns=2;s=Scalar_Static_Double is: 1033833323429888

--- SIMULATE RECONNECT ---

--- RECONNECTED ---
```

You can abort the running application with Ctrl-C.

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 are, especially in the information technology, essential for future success. Concerning our customers, we will constantly be 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