Integration Objects'

Toolkit for OPC UA Client Applications Development

OPC UA Client Toolkit

Version 2.0 Rev.2

QUICK USER GUIDE

OPC UA Client Toolkit Quick User Guide Version 2.0 Rev 2
Published August 2025

ABOUT THIS USER GUIDE

This guide is a quick step by step guide on how to install, configure and run OPC UA Client Toolkit.

INSTALLATION PRE-REQUISITES

In order to properly run any OPC UA Client developed using the OPC UA Client Toolkit, install the following software components on the target system:

- .NET Framework 4.6.1 or higher
- .Net Core 3.1 Runtime or higher

INSTALLING OPC UA CLIENT TOOLKIT

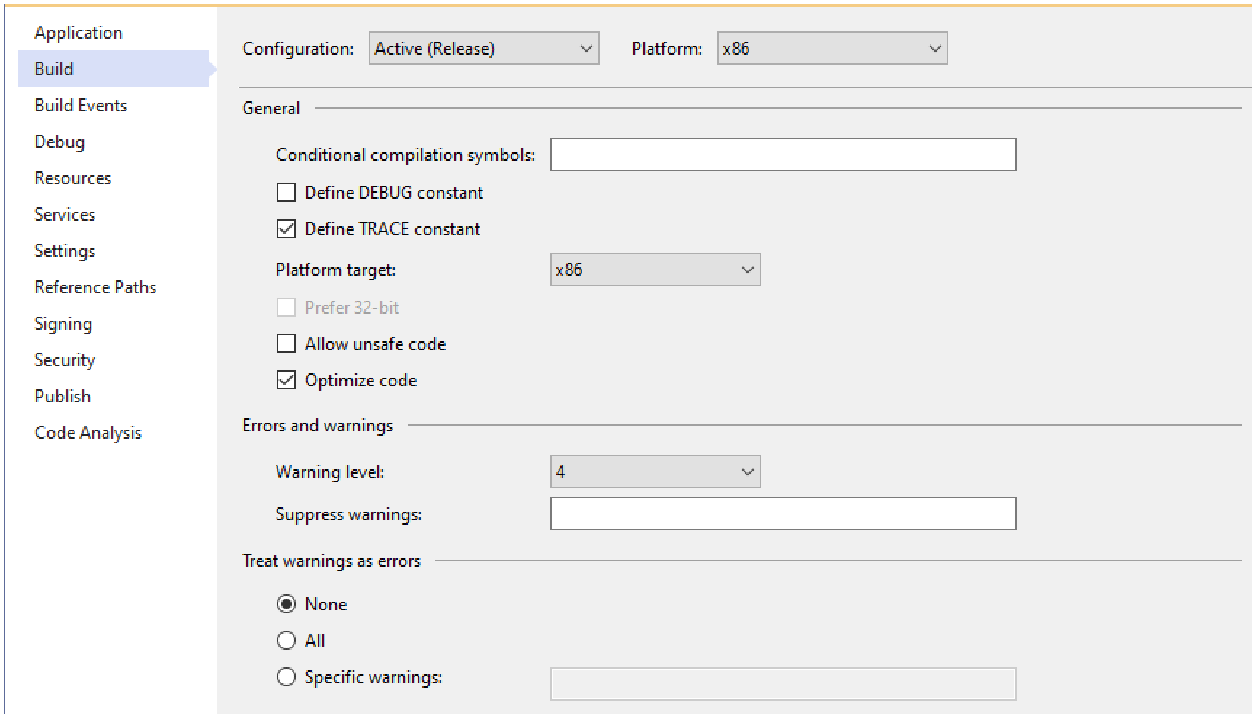
To install the OPC UA Client Toolkit, run the installation executable using an administrator account and the wizard will take you through the different installation steps.

If you are evaluating the OPC UA Client Toolkit, make sure to select demo version in the select features dialog. Otherwise, select full version. The evaluation license allows you to use the toolkit for 30 days and limits the runtime to 2 hours.

COMPILING AND LINKING APPLICATIONS

This section provides the steps on how to compile and correctly link applications to develop a custom OPC UA clients using Integration Objects' OPC UA Client Toolkit with Microsoft Visual Studio 2019.

For users who have to build the application in a **32-bit** machine, the target platform has to be **x86** as illustrated in the screenshot below.



Configuration: Active (Release) Platform: x86

General

Conditional compilation symbols:

□ Define DEBUG constant

☑ Define TRACE constant

Platform target: x86

□ Prefer 32-bit

□ Allow unsafe code

☑ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings:

Treat warnings as errors

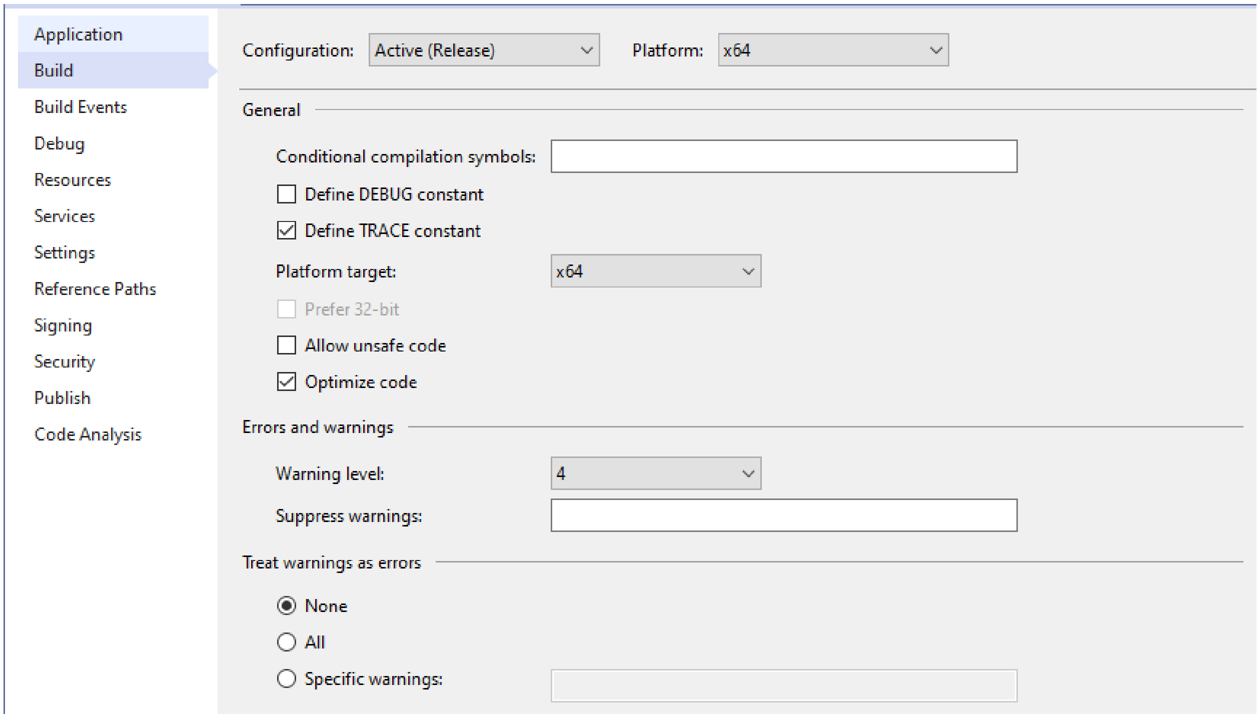
● None

○ All

○ Specific warnings:

Figure 1: Platform Target for 32-bit Machine

For users who have to build the application in a **64-bit** machine, the target platform has to be **Any CPU** as illustrated in the screenshot below.



Configuration: Active (Release) Platform: x64

General

Conditional compilation symbols:

□ Define DEBUG constant

☑ Define TRACE constant

Platform target: x64

□ Prefer 32-bit

□ Allow unsafe code

☑ Optimize code

Errors and warnings

Warning level: 4

Suppress warnings:

Treat warnings as errors

● None

○ All

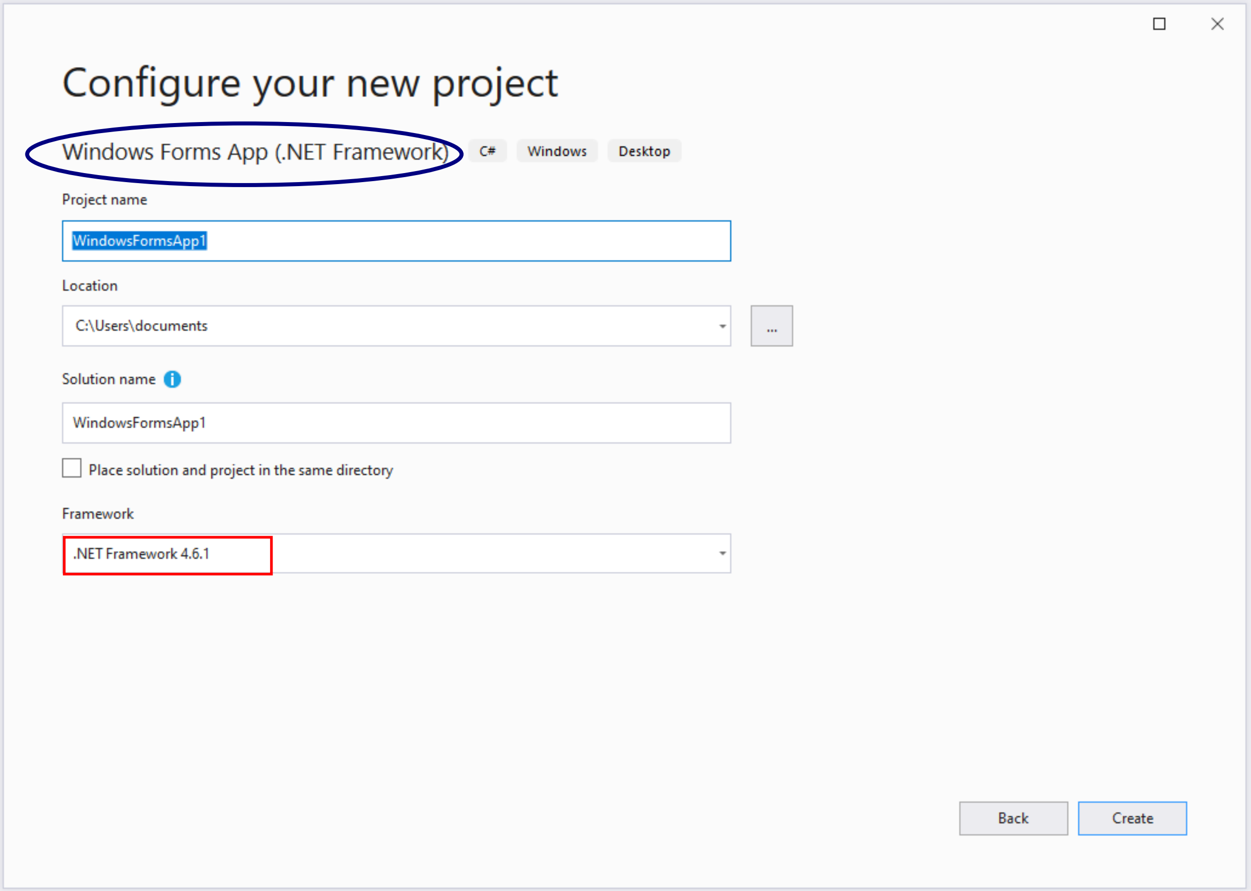
○ Specific warnings:

Figure 2: Platform Target for 64-bit Machine

- ## WINDOWS FORM APPLICATIONS USING .NET FRAMEWORK:

To build a .Net framework OPC UA Client application, follow the steps below:

1. Start Visual Studio 2019 and choose **New Project**. The following window will be displayed.



Configure your new project

Windows Forms App (.NET Framework)

Project name: WindowsFormsApp1

Location: C:\Users\documents

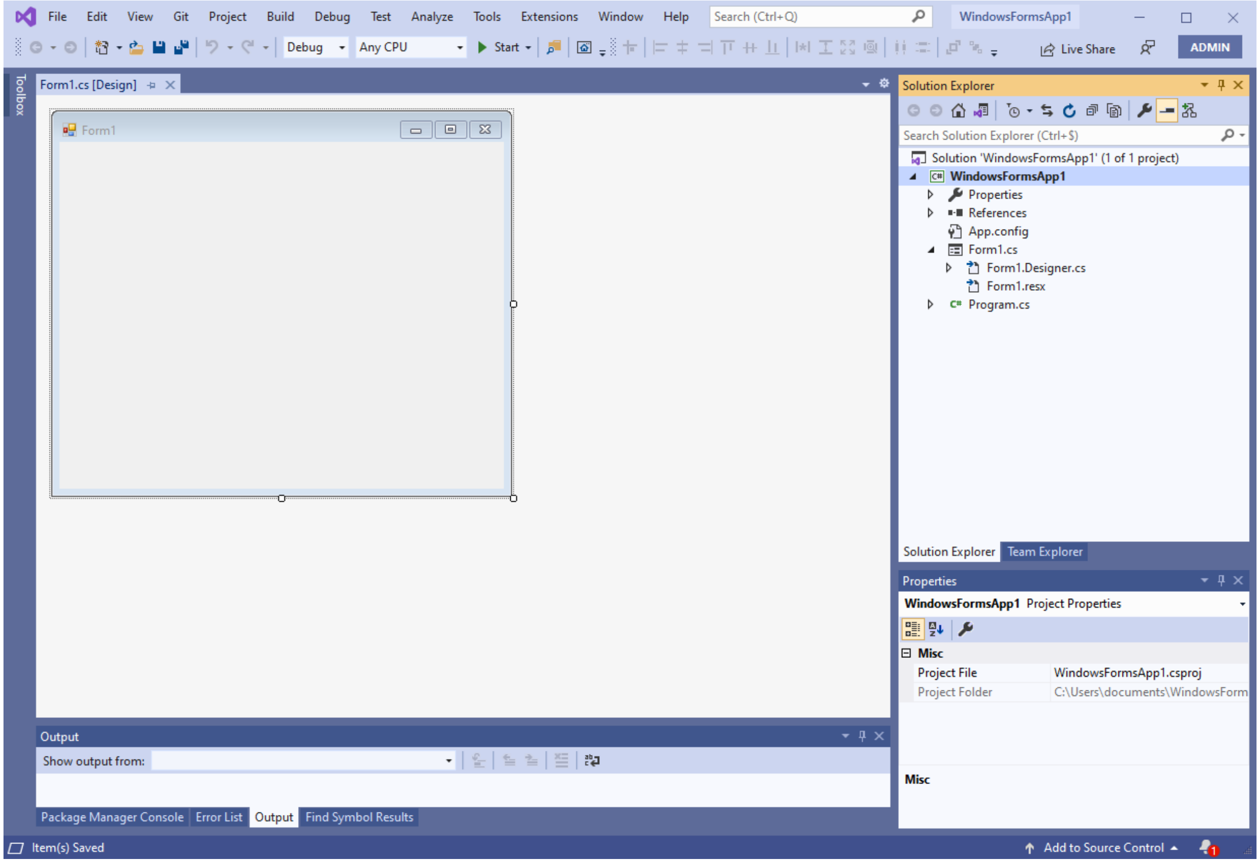
Solution name: WindowsFormsApp1

□ Place solution and project in the same directory

Framework: .NET Framework 4.6.1

Figure 3: New Windows Form Project

2. Choose Visual C# **Windows Forms Application** Project and then click **OK**. A project named WindowsFormsApplication1 with a form called Form1 will be automatically created.



Project File	WindowsFormsApp1.csproj
Project Folder	C:\Users\documents\WindowsForm

Figure 4: Windows Forms Project Template

3. Right click on References then click Add Reference… from the displayed menu.

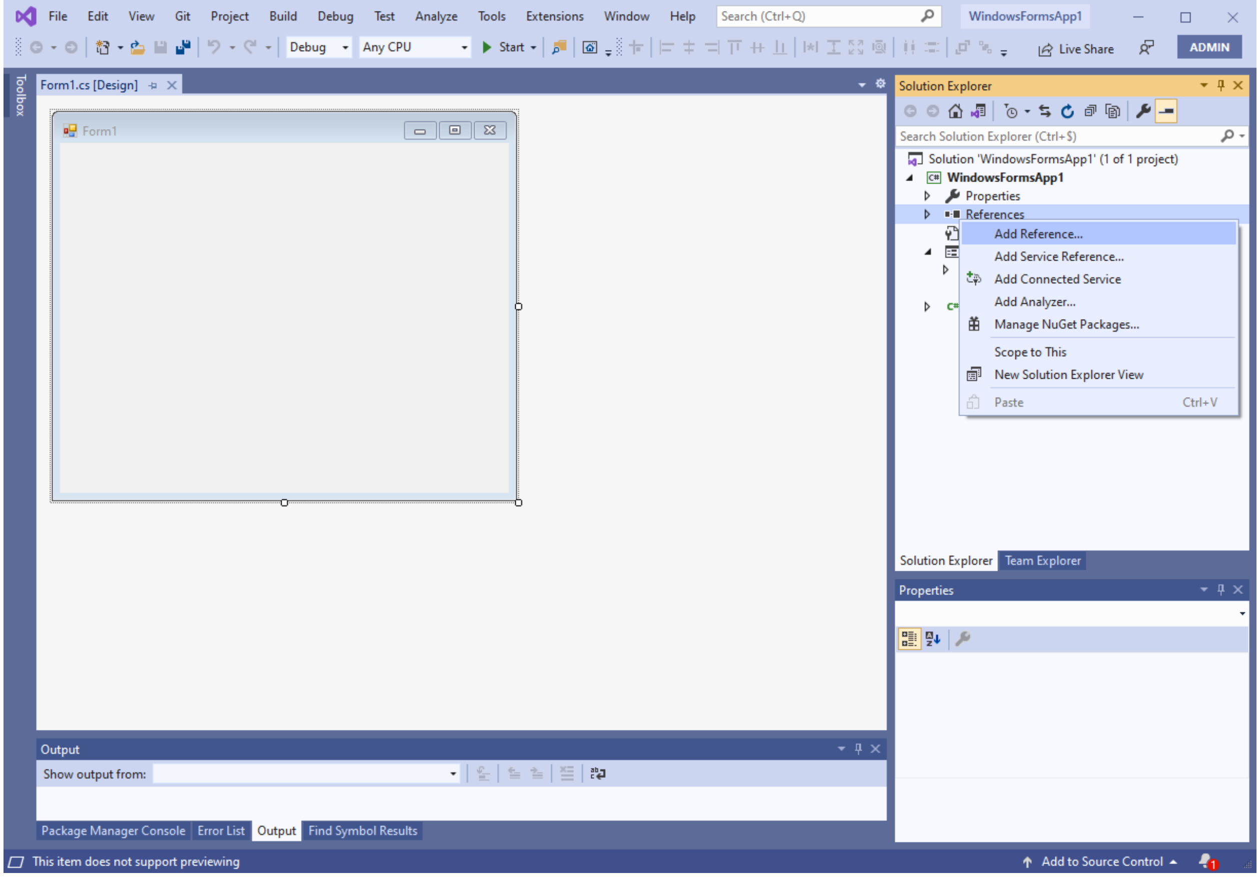
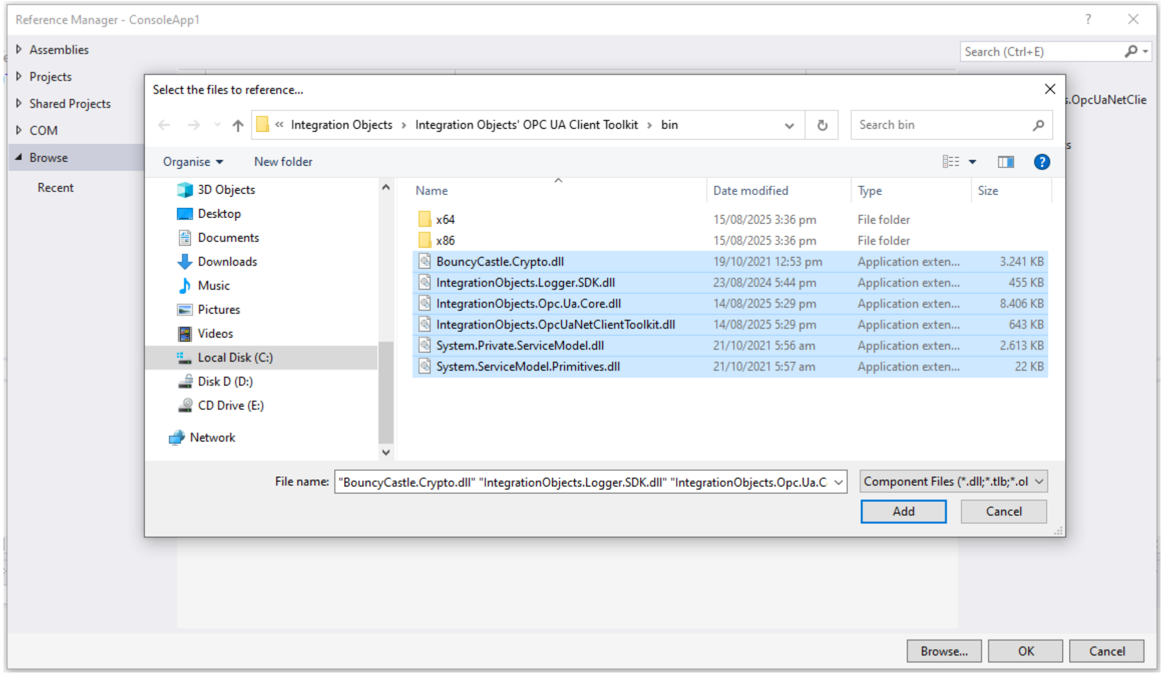


Figure 5: Solution Explorer

4. Select Browse tab from the displayed Add Reference window.

5. Select the following files located under "**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin**":

- IntegrationObjects.OpcUaNetClientToolkit.dll
- IntegrationObjects.Opc.Ua.Core.dll
- IntegrationObjects.Logger.SDK.dll
- BouncyCastle.Crypto.dll
- System.ServiceModel.Primitives.dll
- System.Private.ServiceModel.dll



Name	Date modified	Type	Size
x64	15/08/2025 3:36 pm	File folder	
x86	15/08/2025 3:36 pm	File folder	
BouncyCastle.Crypto.dll	19/10/2021 12:53 pm	Application exten...	3.241 KB
IntegrationObjects.Logger.SDK.dll	23/08/2024 5:44 pm	Application exten...	455 KB
IntegrationObjects.Opc.Ua.Core.dll	14/08/2025 5:29 pm	Application exten...	8.406 KB
IntegrationObjects.OpcUaNetClientToolkit.dll	14/08/2025 5:29 pm	Application exten...	643 KB
System.Private.ServiceModel.dll	21/10/2021 5:56 am	Application exten...	2.613 KB
System.ServiceModel.Primitives.dll	21/10/2021 5:57 am	Application exten...	22 KB

File name: "BouncyCastle.Crypto.dll" "IntegrationObjects.Logger.SDK.dll" "IntegrationObjects.Opc.Ua.C

Figure 6: Choosing a Reference

6. Copy the UA XML configuration file "XXXX.Config.xml" file located under "**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin**" and paste it in the output project.

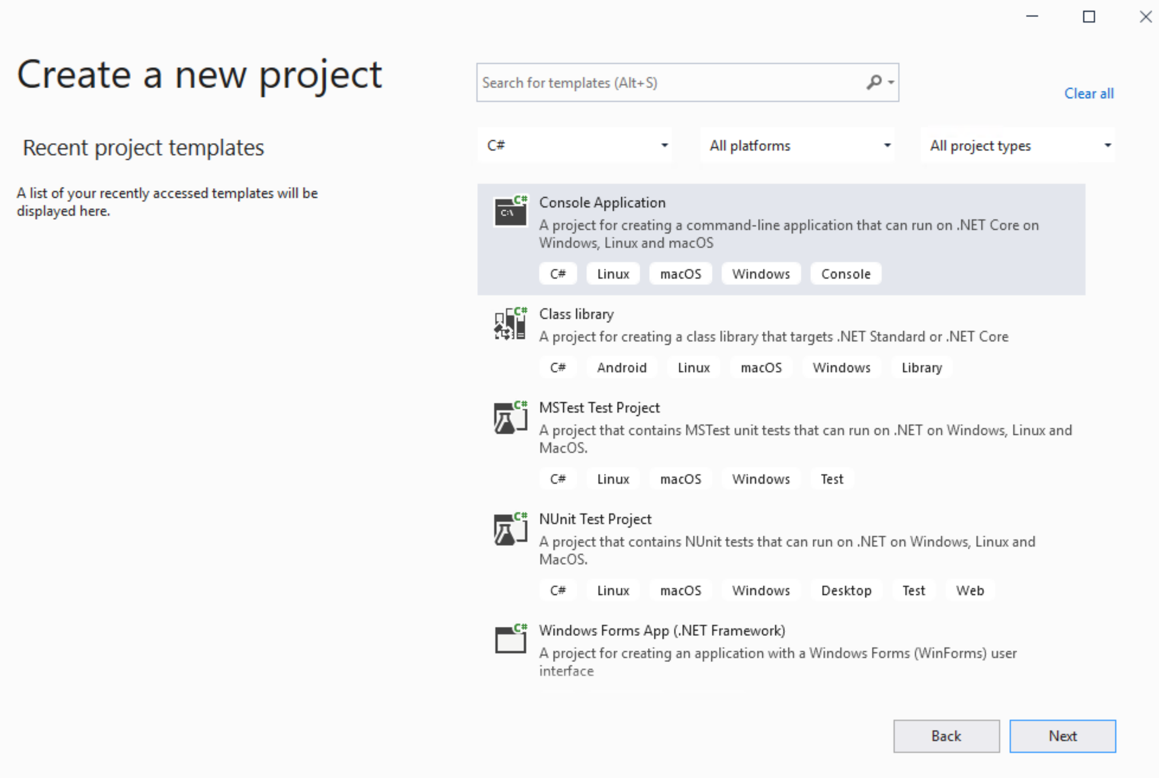
7. Copy the "License.dll" file located under "**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin\x64**" and paste it in the output project.

   Make sure to choose the "License.dll" file located under "**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin\x86**" if you are using the 32-bit version.

• A CONSOLE APPLICATION USING .NET CORE:

To build a .Net Core OPC UA Client application, follow the steps below:

1. Start Visual Studio 2019 and choose **New Project**. The following window will be displayed.



Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

Console Application
A project for creating a command-line application that can run on .NET Core on Windows, Linux and macOS
C# Linux macOS Windows Console

Class library
A project for creating a class library that targets .NET Standard or .NET Core
C# Android Linux macOS Windows Library

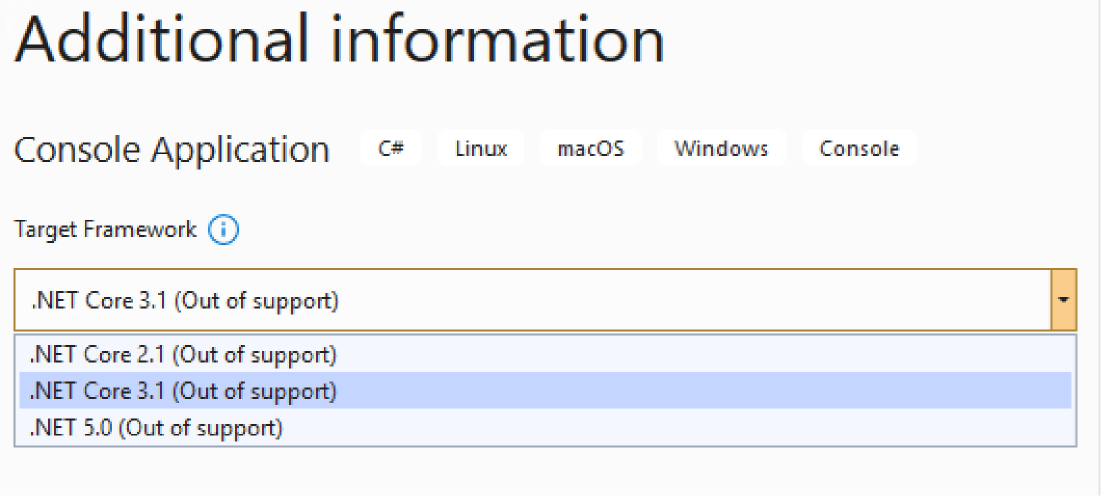
MSTest Test Project
A project that contains MSTest unit tests that can run on .NET on Windows, Linux and MacOS.
C# Linux macOS Windows Test

NUnit Test Project
A project that contains NUnit tests that can run on .NET on Windows, Linux and MacOS.
C# Linux macOS Windows Desktop Test Web

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface

Figure 7: New Console Application Project

2. Choose Visual C# **Console Application** Project and then click **Next**.
3. From Additional information, select the .Net Core version you need to build with your application project and click Create.



Additional information

Console Application

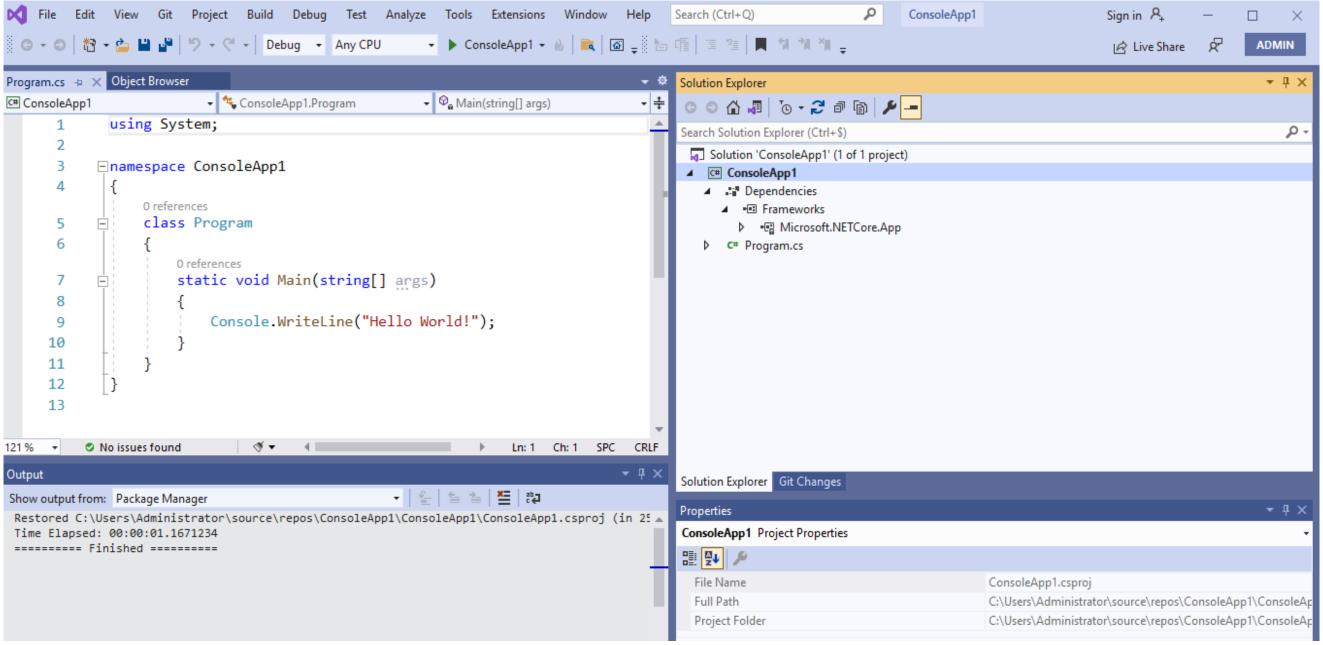
Target Framework

.NET Core 3.1 (Out of support)

.NET Core 2.1 (Out of support)
.NET Core 3.1 (Out of support)
.NET 5.0 (Out of support)

Figure 8: New Console Application Project

A project named ConsoleApp1 will be automatically created.



```
using System;

namespace ConsoleApp1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

```
Restored C:\Users\Administrator\source\repos\ConsoleApp1\ConsoleApp1\ConsoleApp1.csproj (in 25
Time Elapsed: 00:00:01.1671234
========== Finished ==========
```

File Name	ConsoleApp1.csproj
Full Path	C:\Users\Administrator\source\repos\ConsoleApp1\ConsoleAp
Project Folder	C:\Users\Administrator\source\repos\ConsoleApp1\ConsoleAp

Figure 9: Console Application Project Template

4. Right click on **Dependencies** then click Add Project Reference… from the displayed menu.

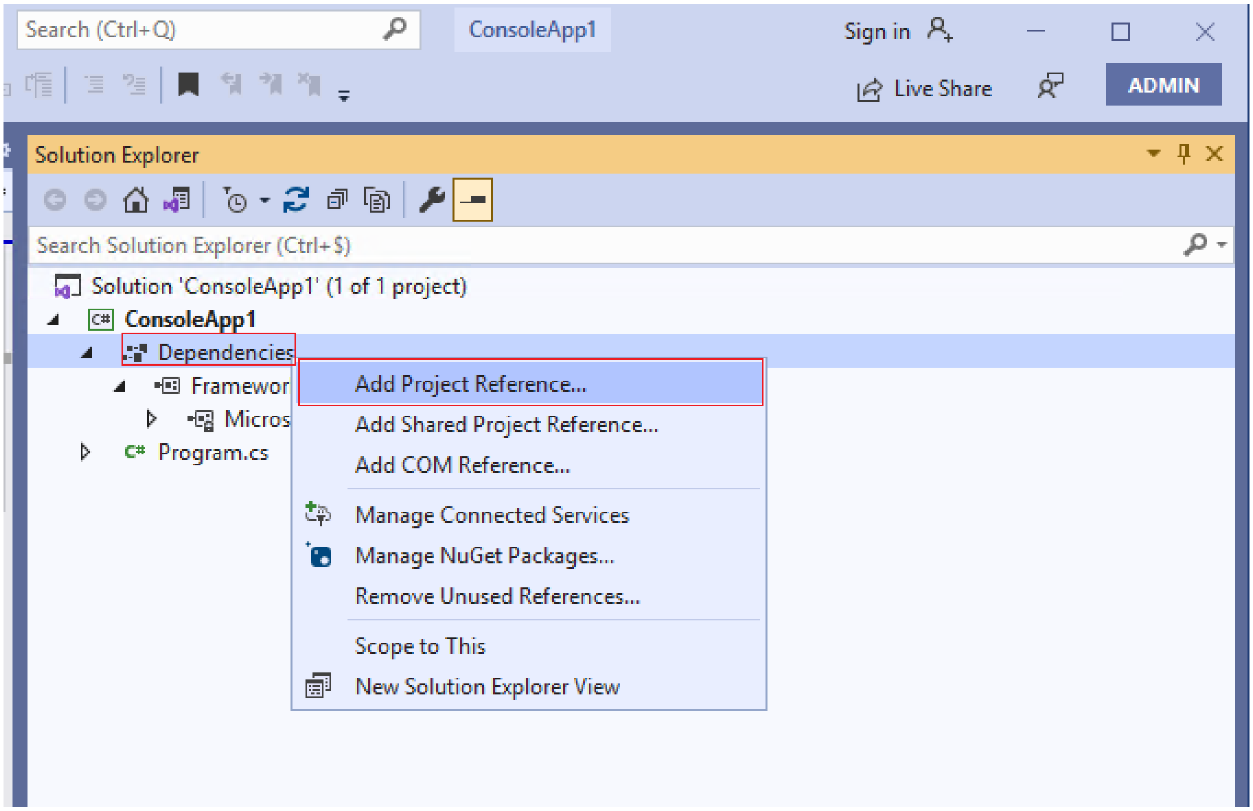
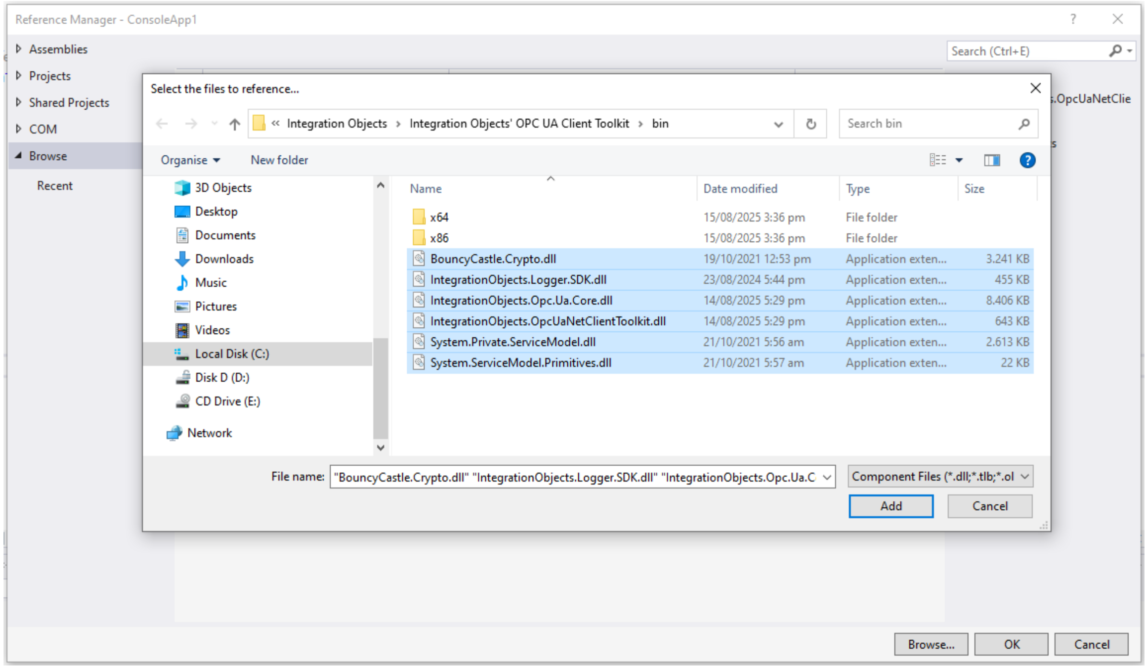


Figure 10: Solution Explorer

5. Select Browse tab from the displayed Add Reference window.

6. Select the following files located under "**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin**":

- IntegrationObjects.OpcUaNetClientToolkit.dll
- IntegrationObjects.Opc.Ua.Core.dll
- IntegrationObjects.Logger.SDK.dll
- BouncyCastle.Crypto.dll
- System.ServiceModel.Primitives.dll
- System.Private.ServiceModel.dll



Name	Date modified	Type	Size
x64	15/08/2025 3:36 pm	File folder	
x86	15/08/2025 3:36 pm	File folder	
BouncyCastle.Crypto.dll	19/10/2021 12:53 pm	Application exten...	3.241 KB
IntegrationObjects.Logger.SDK.dll	23/08/2024 5:44 pm	Application exten...	455 KB
IntegrationObjects.Opc.Ua.Core.dll	14/08/2025 5:29 pm	Application exten...	8.406 KB
IntegrationObjects.OpcUaNetClientToolkit.dll	14/08/2025 5:29 pm	Application exten...	643 KB
System.Private.ServiceModel.dll	21/10/2021 5:56 am	Application exten...	2.613 KB
System.ServiceModel.Primitives.dll	21/10/2021 5:57 am	Application exten...	22 KB

File name: "BouncyCastle.Crypto.dll" "IntegrationObjects.Logger.SDK.dll" "IntegrationObjects.Opc.Ua.C

Component Files (*.dll;*.tlb;*.ol

Figure 11: Choosing a Reference

7. Copy the UA XML configuration file “XXXX.Config.xml” file located under “**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin**” and paste it in the output project.

8. Copy the “License.dll” file located under “**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin\x64**” and paste it in the output project.

Make sure to choose the “License.dll” file located under “**.:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Client Toolkit\bin\x86**” if you are using the 32-bit version.

DEPLOYING A RUNTIME DISTRIBUTION

In order to deploy the client application from the development machine to the runtime machine, follow the steps below:

1. Create a new folder
2. Copy the following files to this new folder:
   - Config.json
   - License.dll
   - IntegrationObjects.Logger.SDK.dll
   - IntegrationObjects.Opc.Ua.Core.dll
   - IntegrationObjects.OpcUaNetClientToolkit.dll
   - The application executable and any other custom depending assembly
   - The UA XML configuration file (XXXX.Config.xml, where XXXX is the name of your OPC UA client application)
   - BouncyCastle.Crypto.dll
   - System.ServiceModel.Primitives.dll
   - System.Private.ServiceModel.dll
   - ConnectionConfig.json (when using the OPC UA .Net Core Toolkit)

3. Move the folder to the runtime machine

Make sure that OPC UA Client Toolkit is not installed in the runtime machine and that the path of the application folder does not include the words “Debug” or “Release”.

For additional information on this guide, questions or problems to report, please contact:

Offices

- Americas:	+1 713 609 9208
- Europe-Africa-Middle East:	+216 71 195 360

Email

- Support Services: customerservice@integrationobjects.com
- Sales: sales@integrationobjects.com

To find out how you can benefit from other Integration Objects products and custom-designed solutions, please visit us on the Internet:

Online

- www.integrationobjects.com