

Integration Objects' .NET Toolkit for OPC DA/HDA/A&E Client Applications Development

OPC .NET Client Toolkit Version 3.0 Rev.0

QUICK USER GUIDE



OPC .NET Client Toolkit Quick User Guide Version 3.0 Rev 0 Published December 2023 Copyright © 2009-2023 Integration Objects. All rights reserved.



ABOUT THIS USER GUIDE

This guide is a step by step guide that lists the main steps on how to install, configure and run OPC .Net Client Toolkit.

INSTALLATION PRE-REQUISITES

In order to properly run any OPC Client developed by using the OPC .NET Client Toolkit, install these software components on the target system:

- The OPC core components 3.00 which consist of all shared OPC modules including the DCOM proxy/stub libraries, the OPC Server Enumerator, .NET wrappers, etc. You can alternatively apply the OPC Core Components 3.00 Redistributable delivered with the current package or download it from the OPC Foundation site (www.opcfoundation.org).
- .NET Framework 4.6.1 or higher
- .NET Core 3.1 or higher.

INSTALLING OPC .NET CLIENT TOOLKIT

To install the OPC .NET 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 .NET Client Toolkit, make sure to select demo version in the setup type 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 clients using Integration Objects' OPC Client .NET Toolkit with Microsoft Visual Studio 2017.

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.



Application Build	Configuration: Release V Platform: x86 V					
Build Events	General					
Debug Resources Settings Reference Paths Signing Security Publish Code Analysis	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: 3					
	O None					
	O All O Specific warnings: Output Output					
	Output path: bin/Release/ Browse XML documentation file:					
	Advanced					

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.

Application Build	Configuration: Release V Platform: Any CPU V
Build Events	General
Debug Resources Services Settings Reference Paths Signing Security Publish	Conditional compilation symbols: Define DEBUG constant Define TRACE constant Platform target: Any CPU Prefer 32-bit Allow unsafe code Optimize code
Code Analysis	Errors and warnings 3 Warning level: 3 Suppress warnings:

Figure 2: Platform Target for 64-bit Machine



1.1. Step 1

Start Visual Studio 2017 and choose New Project. The following window will be displayed.

New Project							?	×
Recent		Sort by:	Default		Search (Ctrl+E)			، م
▲ Installed		C#	WPF App (.NET Framework)	Visual C#	Type: Visual (C#		
 Visual C# Visual Basic Visual C++ 			Windows Forms App (.NET Framework)	Visual C#	A project for creating an a Windows Forms user inter		pplication with face	
 Visual C++ Windows Desktop Cross Platform MFC/ATL Test Other InstallShield Projects Other Project Types 		<u> </u>	Console App (.NET Framework)	Visual C#				
			Class Library (.NET Standard)	Visual C#				
			Class Library (.NET Framework)	Visual C#				
		C# ⊒	Shared Project	Visual C#				
◊ Online			Class Library (Legacy Portable)	Visual C#				
Not finding what y	ou are looking for?							
Name	WindowsFormsApp1							
Location:	C:\Llsers\ssaidi\sourc	e\renos		•	Browse			
Solution	Create new solution	Croste new solution			browse			
Solution.	WindowsFormsApp1				Contrational			
Solution name:	NET Example 4.6				Add to Source	ry for solution	1	
Framework:	INET Framework 4.0.	*				Control		
						OK	Can	cel

Figure 3: New Windows Forms Project

Choose Visual C# Windows Forms Application Project and then click OK.



1.2. Step 2

A project named WindowsFormsApp1 with a form called Form1 will be automatically created.



Figure 4: Windows Forms Project Template



1.3. Step 3

Add reference to the OPC .NET Client Toolkit as shown below.

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



Figure 5: Solution Explorer

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



Reference Manager - Window	sFormsApp1		? ×
Assemblies	Targeting: .NET Framework 4.6.1		Search (Ctrl+E)
 Framework Extensions Projects Shared Projects COM Browse 	Name Accessibility CustomMarshalers ISymWrapper Microsoft.Activities.Build Microsoft.Build.Conversion.v4.0 Microsoft.Build.Framework Microsoft.Build.Framework Microsoft.Build.Tramework Microsoft.Build.Tramework Microsoft.Build.Tramework Microsoft.Build.Tramework Microsoft.Build.Tramework Microsoft.Script Microsoft.VisualBasic Microsoft.VisualBasic.Compatibility Microsoft.VisualBasic.Compatibility.Data Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC Microsoft.VisualC PresentationFramework PresentationFramework PresentationFramework PresentationFramework.Aero PresentationFramework.Aero2 PresentationFramework.Aero2 PresentationFramework.Aero2 PresentationFramework.Aero2 PresentationFramework.Aero2 PresentationFram	Version 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 4.0.0.0 10.0.0.0 10.0.0.0 10.0.0.0 10.0.0.0 10.0.0.0 10.0.0.0 4.0.0.	Name: Accessibility Created by: Microsoft Corporation Version: 4.0.0.0 File Version: 4.6.1055.0 built by: NETFXREL2
		Brows	e OK Cancel

Figure 6: Adding a Reference

3. Select IntegrationObjects.OPCNetClientSDK.dll located under: .:\Program Files\Integration Objects\Integration Objects' OPC .NET Client Toolkit\bin



Select the files to referen	ice					×
← → • ↑ <mark> </mark> «	Integra	tion Objects' OPC .NET Client Toolkit > bin	~	Ō	Search bin	م
Organize 👻 New	folder					• 🔳 🕐
 Quick access Desktop Downloads Documents Pictures HDAServerToolk Local Disk (C:) 	* * * * *	Name x64 x86 IntegrationObjects.Logger.SDK.dll IntegrationObjects.OPCNetClientSDK.dll OpcComRcw.dll	Date modified 2023-12-13 09:12 2023-12-13 09:23 2022-11-09 09:25 2023-11-01 15:25 2022-11-09 09:25		Type File folder File folder Application exten Application exten	Size 319 KB 695 KB 86 KB
	~					
F	ile name:	IntegrationObjects.OPCNetClientSDK.dll		~	Component Files (* Add	.dll;*.tlb;*.ol ∨ Cancel

Figure 7: Choosing a Reference



If OPC .NET Client Toolkit license changed to full license, make sure to reference the new full version .dll files on the application sample project or to copy them under the output folder.

DEPLOYING A RUNTIME DISTRIBUTION

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

- 1. Create a new folder
- 2. Copy the following files to this new folder:
 - ConfigOPCClientSDK.ini
 - IntegrationObjects.Logger.SDK.dll
 - IntegrationObjects.OPCNetClientSDK.dll
 - License.dll
 - OpcComRcw.dll
 - The application executable and any other custom depending assembly
- 3. Move the folder to the runtime machine



Make sure that OPC .NET 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: <u>sales@integrationobjects.com</u>

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