



OPC UA SDK für .NET

Getestet? Du willst es?

[Lizenzmodell](#) [Preise](#) [Angebot](#) [Jetzt bestellen](#)

[Buch - Die gesamte Anleitung als eBook](#)

Development Guides

[Client Development Guide](#) [Server Development Guide](#) [Anwendungsfälle](#) [Häufige Fragen](#)

Download

Das OPC UA .NET SDK kommt mit einer **Testlizenz die je Anwendungsstart 30 Minuten uneingeschränkt zur Client- und Serverentwicklung** verwendet werden kann. Sollte diese Einschränkung ihre Evaluationsmöglichkeiten einschränken, besteht die Möglichkeit eine **alternative Testlizenz** bei uns **kostenlos** zu beantragen. Fragen Sie einfach unseren Support (via support@traeger.de) oder lassen Sie sich gleich direkt von uns beraten und offene Fragen durch unsere Entwickler klären!

OPC UA .NET SDK für Clients - Evaluationspaket¹⁾

[Download ZIP Archiv von Opc.UaFx.Client](#) (Version: 2.42.3.0 - 2024-09-11)

[Download NuGet Paket von Opc.UaFx.Client](#) (Version: 2.42.3.0 - 2024-09-11)

OPC UA .NET SDK für Clients und Server - Evaluationspaket²⁾

[Download ZIP Archiv von Opc.UaFx.Advanced](#) (Version: 2.42.3.0 - 2024-09-11)

[Download NuGet Paket von Opc.UaFx.Advanced](#) (Version: 2.42.3.0 - 2024-09-11)

OPC UA .NET SDK für LabVIEW Clients - Evaluationspaket³⁾

[Download ZIP Archiv von Opc.UaFx.Client.LabView](#) (Version: 1.1.3.0 - 2024-06-14)

OPC UA .NET SDK für Unity Clients - Evaluationspaket⁴⁾

OPC UA .NET SDK für .NET Framework 3.5 Clients und Server - Evaluationspaket⁵⁾

[Download ZIP Archiv von Opc.UaFx.Advanced](#) (Version: 2.0.1.1 - 2017-06-07)

[Download NuGet Paket von Opc.UaFx.Advanced](#) (Version: 2.0.1.1 - 2017-06-07)

[OPC Watch](#) (Version: 2.42.1.0 - 2024-07-25)

Ein kostenloser und einfacher, aber professioneller OPC UA Client für den Zugriff auf OPC UA Server.

[Versionshistorie - Die Liste der Verbesserungen pro Version](#)

Preview Download

OPC UA .NET SDK für Clients - PREV1 für OPC UA v1.04⁶⁾

OPC UA .NET SDK für Clients und Server - PREV1 für OPC UA v1.04⁷⁾

Runtime Download

Zur Anbindung von **OPC Classic** Servern in 64 Bit-Anwendungen und zur Enumeration (= Discovery) von lokalen OPC Classic Servern müssen die Core Components von der OPC Foundation auf dem Zielsystem installiert sein. Diese finden sie unter anderen auf der [Seite der OPC Foundation](#) oder auch hier:

OPC Core Components Redistributables⁸⁾

📄 [Download ZIP Archiv von OPC Core Components](#) (Version: 3.00.107 - 2018-01-30)

📄 [Download ZIP Archiv von OPC Core Components](#) (Version: 3.00.108 - 2019-12-20)

OPC UA Client

[OPC UA Client Development Guide](#)

Beispiel Code: OPC UA Client

- C#
- VB

```
namespace Client
{
    using System;
    using System.Threading;

    using Opc.UaFx.Client;

    public class Program
    {
        public static void Main()
        {
            using (var client = new OpcClient("opc.tcp://localhost:4840")) {
                client.Connect();

                while (true) {
                    var temperature = client.ReadNode("ns=2;s=Temperature");
                    Console.WriteLine("Current Temperature is {0} °C", temperature);

                    Thread.Sleep(1000);
                }
            }
        }
    }
}
```

```
Imports System
Imports System.Threading

Imports Opc.UaFx.Client

Namespace Client
    Public Class Program
        Public Shared Sub Main()
            Using client = New OpcClient("opc.tcp://localhost:4840")
                client.Connect()

                While True
                    Dim temperature = client.ReadNode("ns=2;s=Temperature")
                    Console.WriteLine("Current Temperature is {0} °C", temperature)

                    Thread.Sleep(1000)
                End While
            End Using
        End Sub
    End Class
End Namespace
```

OPC UA Server

[OPC UA Server Development Guide](#)

Beispiel Code: OPC UA Server

- [C#](#)
- [VB](#)

```
namespace Server
{
    using System.Threading;

    using Opc.UaFx;
    using Opc.UaFx.Server;

    public class Program
    {
        public static void Main()
        {
            var temperatureNode = new OpcDataVariableNode<double>("Temperature", 100.0);

            using (var server = new OpcServer("opc.tcp://localhost:4840/", temperatureNode))
            {
                server.Start();

                while (true) {
                    if (temperatureNode.Value == 110)
                        temperatureNode.Value = 100;
                    else
                        temperatureNode.Value++;

                    temperatureNode.ApplyChanges(server.SystemContext);
                    Thread.Sleep(1000);
                }
            }
        }
    }
}
```

```
Imports System.Threading

Imports Opc.UaFx
Imports Opc.UaFx.Server

Namespace Server
    Public Class Program
        Public Shared Sub Main()
            Dim temperatureNode = New OpcDataVariableNode(Of Double)("Temperature", 100.0)

            Using server = New OpcServer("opc.tcp://localhost:4840/", temperatureNode)
                server.Start()

                While True
                    If (temperatureNode.Value = 110) Then
                        temperatureNode.Value = 100
                    Else
                        temperatureNode.Value += 1
                    End If

                    temperatureNode.ApplyChanges(server.SystemContext)
                    Thread.Sleep(1000)
                End While
            End Using
        End Sub
    End Class
End Namespace
```

- 1) , 2) , 3) , 4) , 5) Mit Ihrem „License Code“ wird das Paket zur produktiven Vollversion.
6) , 7) Nicht für den produktiven Einsatz empfohlen.
8) Bereitgestellt durch die OPC Foundation

Inhaltsverzeichnis

Getestet? Du willst es?	1
Development Guides	2
Download	2
Preview Download	2
Runtime Download	3
OPC UA Client	3
Beispiel Code: OPC UA Client	3
OPC UA Server	4
Beispiel Code: OPC UA Server	4

