

# OPC UA SDK for .NET



**Successful with only a few lines of code**

[Book - The whole Manual as eBook](#)

## Tested? You want it?

[License Model Prices Quotation Order Now](#)

# Development Guides

[Client Development Guide](#) [Server Development Guide](#)

## Class Library

[Opc.UaFx Namespace](#) [Opc.UaFx.Client Namespace](#) [Opc.UaFx.Server Namespace](#)

## Download

**OPC UA .NET SDK for Clients** - Evaluation Package<sup>1)</sup>

[Download ZIP Archive of Opc.UaFx.Client](#) (Version: 2.8.2.0 - 2019-11-06)

[Download NuGet Package of Opc.UaFx.Client](#) (Version: 2.8.2.0 - 2019-11-06)

**OPC UA .NET SDK for Clients and Servers** - Evaluation Package<sup>2)</sup>

[Download ZIP Archive of Opc.UaFx.Advanced](#) (Version: 2.8.2.0 - 2019-11-06)

[Download NuGet Package of Opc.UaFx.Advanced](#) (Version: 2.8.2.0 - 2019-11-06)

**OPC UA .NET SDK for Unity Clients** - Evaluation Package<sup>3)</sup>

[Download ZIP Archive of Opc.UaFx.Client](#) (Version: 2.7.2.0 - 2019-05-10)

[Download NuGet Package of Opc.UaFx.Client](#) (Version: 2.7.2.0 - 2019-05-10)

[OPC Watch](#) (Version: 2.8.2.0 - 2019-11-06)

A free and simple but professional OPC UA Client to access OPC UA Servers.

[Version History](#) - The list of improvements in each version

## Preview Download

**OPC UA .NET SDK for Clients and Servers PREV1 for ...**<sup>4)</sup>

- Structured Datatypes (Server-side)
- NodeSets (Companion Specifications like UMATI)

[Download ZIP Archive of Opc.UaFx.Advanced](#) (Version: 2.9.0.0 - 25.11.2019)

[Download NuGet Package of Opc.UaFx.Advanced](#) (Version: 2.9.0.0 - 25.11.2019)

## OPC UA Client

[OPC UA Client Development Guide](#)

[Example C# Code OPC UA Client](#)

```
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);
                }
            }
        }
    }
}
```

# OPC UA Server

[OPC UA Server Development Guide](#)

[Example C# Code OPC UA Server](#)

```
namespace Server
{
    using System.Threading;

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

    internal static 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);
                }
            }
        }
    }
}
```

# General

## Terms

### OPC UA

OPC UA stands for OPC Unified Architecture, shortened OPC UA. Contrasting to the predecessor OPC, OPC UA especially differentiates itself through the ability to not only transport machine data (measurements, parameters etc.), but also describe the data semantically in order for machines to read it. OPC UA means: **O**peness **P**roductivity **C**onnectivity **U**nified **A**rchitecture.

### Node

The "Node" is the most basic element of the OPC UA. Nearly every element is "reduced" to one "Node", so to say. Hearby the Nodes stand within direct relation to each other.

The Wikipedia definition about the OPC Unified Architecture contains a fitting description for the term "Node":

***"The OPC information model is a so-called Full Mesh Network based on nodes. Nodes hold process data as well as all other types of metadata."*** Source:

[wikipedia.org/wiki/OPC\\_Unified\\_Architecture](https://wikipedia.org/wiki/OPC_Unified_Architecture)

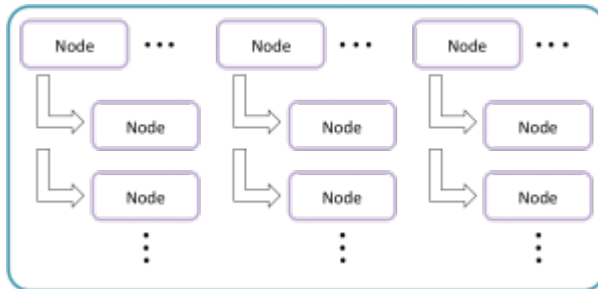
- A Node resembles an object from object oriented programming.
- A Node has attributes, which can be read (Data Access (DA), Historical Data Access (HDA)).
- Nodes are used for process data as well as for all other types of metadata.
- The therefore modelled OPC Address Space contains a type model with which all data types are

specified.

## NodeId

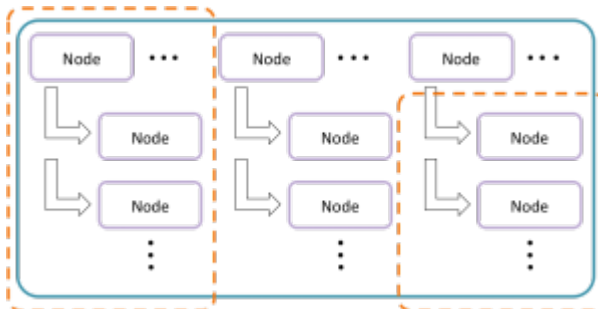
**Node** **ID** The OPC specification defines that every Node can be uniquely identified in the Address Space via an Identifier (= **NodeId**). The **NodeId** is defined either by a GUID (Global Unique Identifier), a numeric expression, an array of bytes or a string value. In general but not necessarily, the NodeId contains the "**Namespace**".

## Address Space



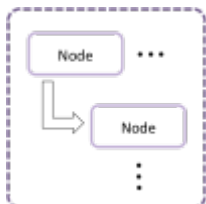
All **Nodes** supplied and processed in the OPC UA are administrated in a so-called **Address Space**. The **Address Space** depicts a kind of logical storage. In this "storage" the contained **Nodes** can logically refer to one or more **Nodes** in the same or another **Address Space**.

## View



The "Address Space" mentioned / visualized earlier can be logically segmented into one or more Views. While there is one **Default View**, **Custom Views** can contain one or more Nodes.

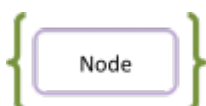
## NodeManager



The **Node Manager** supplies one or more Nodes and defines their relationships towards each other. Predefined **System Node Managers** are:

- Core Node Manager (defines i.a. Type Nodes and System Nodes)
- Diagnostics Node Manager (supplies Nodes for diagnostics)
- Master Node Manager (the "administrator" of all Node Managers, it delegates calls to the concerning Node Managers)

## Service



OPC UA defines a series of different **Services** by means of which the Client interacts with the Server. Those **Services** are server-sided implemented as Methods and are used for:

- reading and writing Node attributes / values
- administrating Node References

- browsing of Nodes
- reading and writing historical values
- calling Methods
- administrating subscriptions
- e.a.

<sup>1)</sup> , <sup>2)</sup> , <sup>3)</sup> Your "License Code" turns the package into a productive full version.

<sup>4)</sup> Not recommended for productive use.

# Table of Contents

<b>Tested? You want it?</b> .....	1
<b>Development Guides</b> .....	2
<b>Class Library</b> .....	2
<b>Download</b> .....	2
<b>Preview Download</b> .....	2
<b>OPC UA Client</b> .....	2
Example C# Code OPC UA Client .....	2
<b>OPC UA Server</b> .....	3
Excample C# Code OPC UA Server .....	3
<b>General</b> .....	4
Terms .....	4
OPC UA .....	4
Node .....	4
NodeId .....	5
Address Space .....	5
View .....	5
NodeManager .....	5
Service .....	5

