

<https://www.halvorsen.blog>



OPC UA with Visual Studio and C#

Using “OPC UA Server Simulator” and “OPC UA .NET SDK”

Hans-Petter Halvorsen

Contents

- Introduction to OPC
 - Communication Protocol for Data Exchange between Devices from different Manufacturers typically used in Industrial and Automation Systems
- OPC UA
 - The Next Generation OPC. Cross-platform. Works with IoT/IIoT
- OPC UA Server Simulator
 - Free OPC UA Server that can be used for Testing and Education
- “OPC UA Client” Tool
 - Free OPC UA Client that can be used for Testing and Education
- OPC UA .NET SDK
 - Free Evaluation license which can be used unlimited for each application but runs only for 30 minutes before restart is required
 - Visual Studio/C# Example
 - Improved Example

<https://www.halvorsen.blog>



OPC

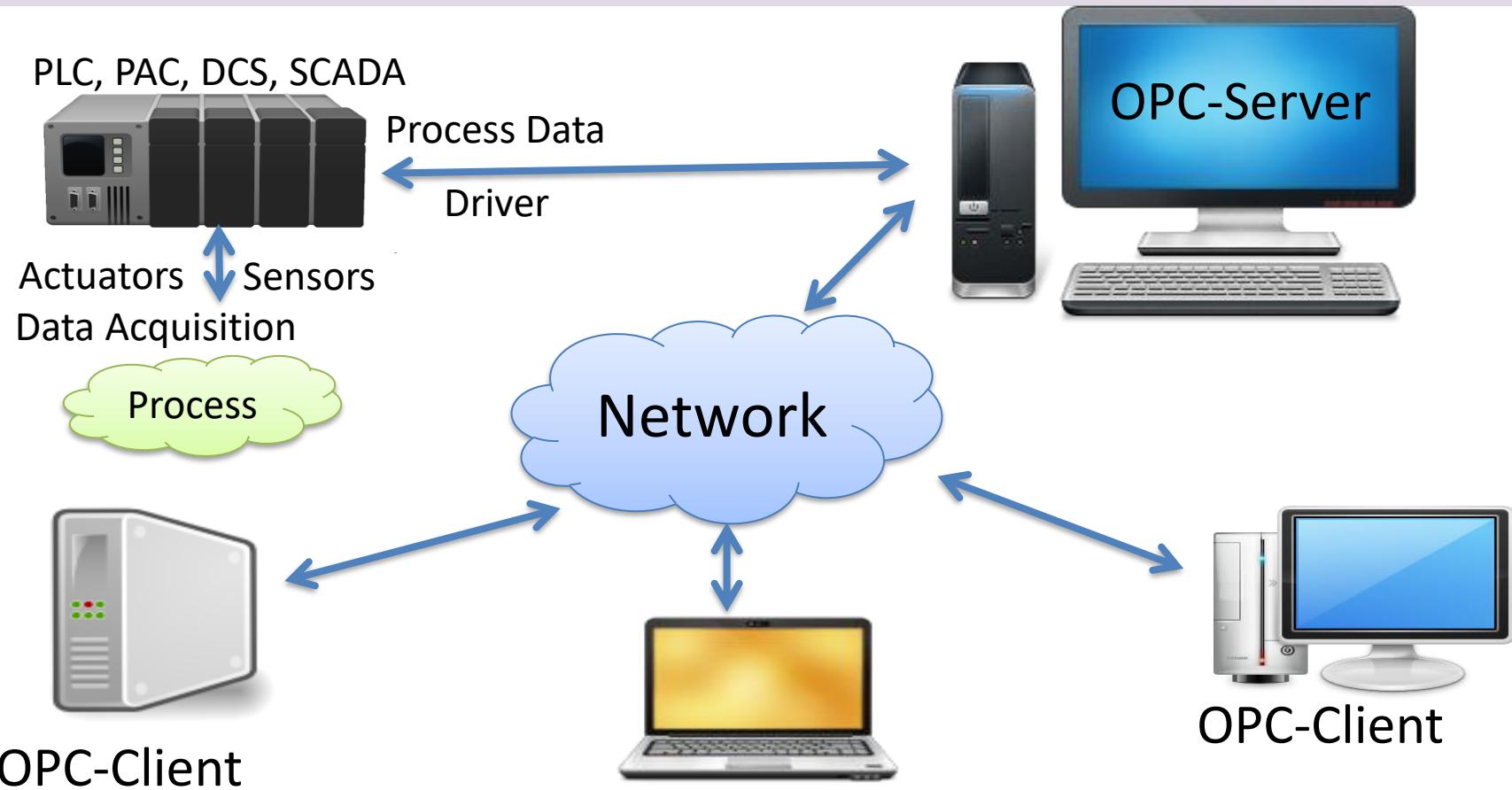
Hans-Petter Halvorsen

[Table of Contents](#)

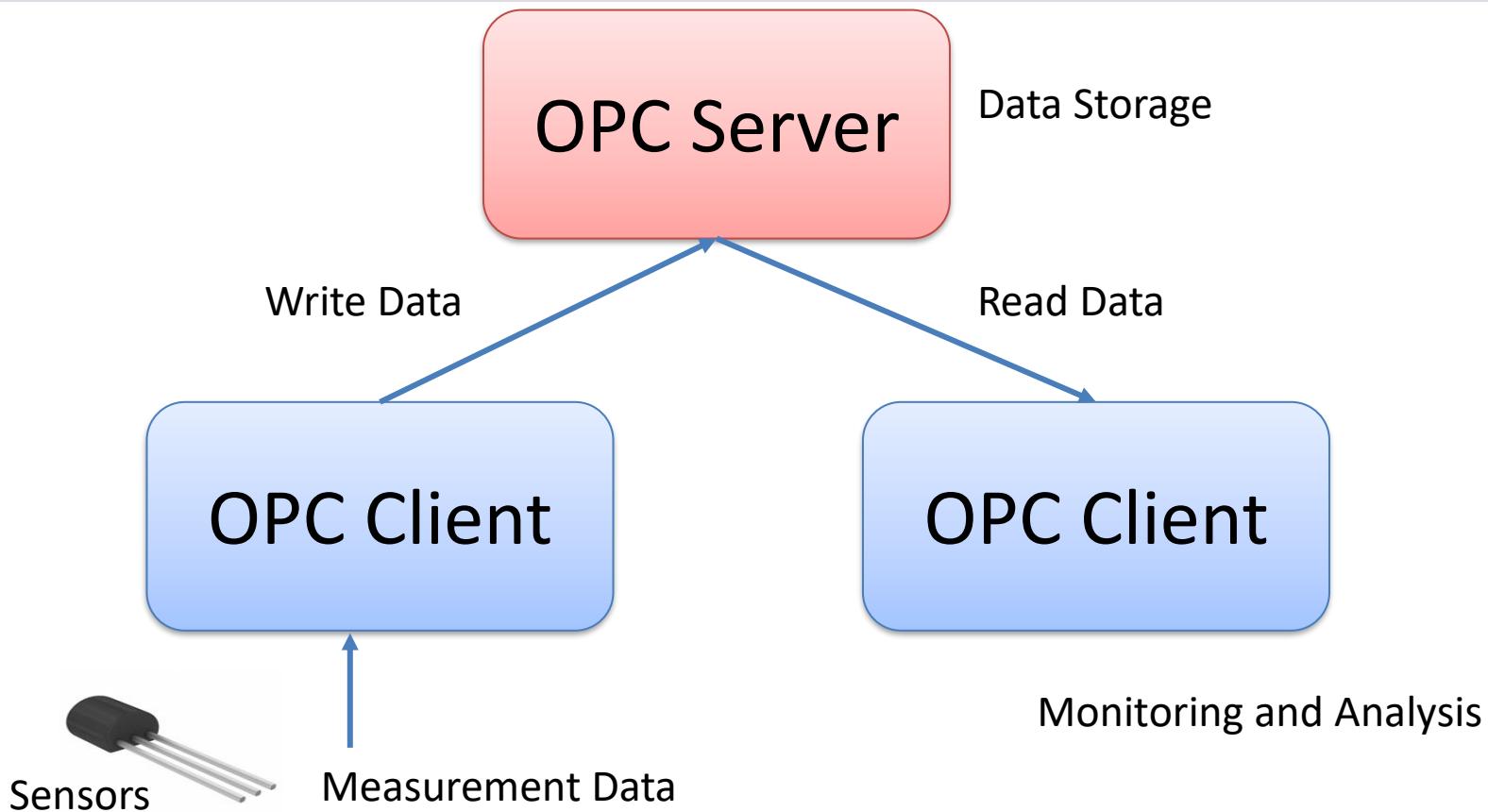
What is OPC?

- A standard that defines the communication of data between devices from different manufactures
- Requires an **OPC server** that communicates with the **OPC clients**
- OPC allows “plug-and-play”, gives benefits as reduces installation time and the opportunity to choose products from different manufactures
- Different standards: “Real-time” data (**OPC DA**), Historical data (**OPC HDA**), Alarm & Event data (**OPC AE**), etc.

Typical OPC Scenario



OPC Server and Client(s)



<https://www.halvorsen.blog>



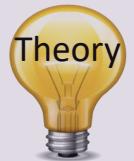
OPC UA

Hans-Petter Halvorsen

[Table of Contents](#)

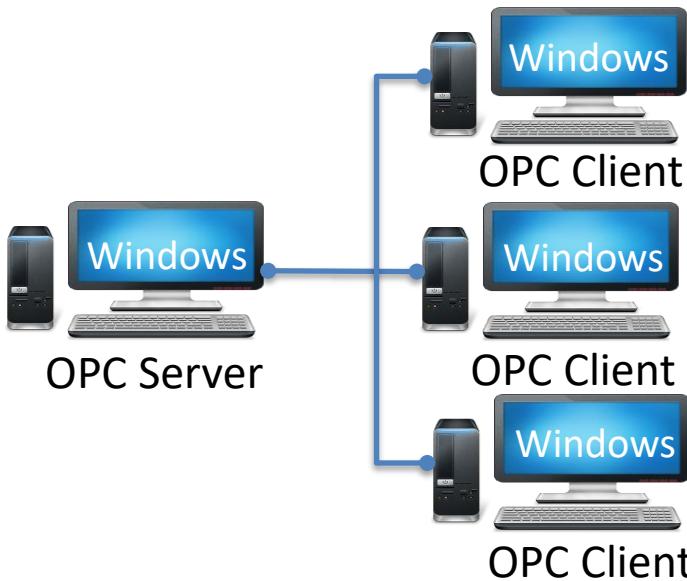
OPC UA

- UA – Unified Architecture
- The Next Generation OPC
- Cross Platform. “Classic” OPC works only for Windows
- Based on Modern Software/Network Architecture (No DCOM problems!)
- It makes it easier to transmit and receive data in a modern data network/Internet



Classic OPC vs. OPC UA

Classic OPC (DCOM)



OPC UA

The server (or clients) can be an embedded system, LINUX, Windows, etc.



OPC UA Server



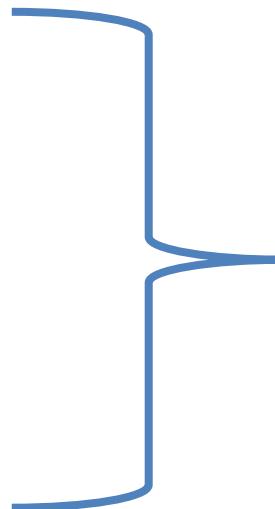
Classic OPC requires a Microsoft Windows operating system to implement COM/DCOM server functionality. By utilizing SOA and Web Services, OPC UA is a platform-independent system that eliminates the previous dependency on a Windows operating system. By utilizing SOAP/XML over HTTP, OPC UA can deploy on a variety of embedded systems regardless of whether the system is a general purpose operating system, such as Windows, or a deterministic real-time operating system.

<http://www.ni.com/white-paper/13843/en/>

OPC Specifications

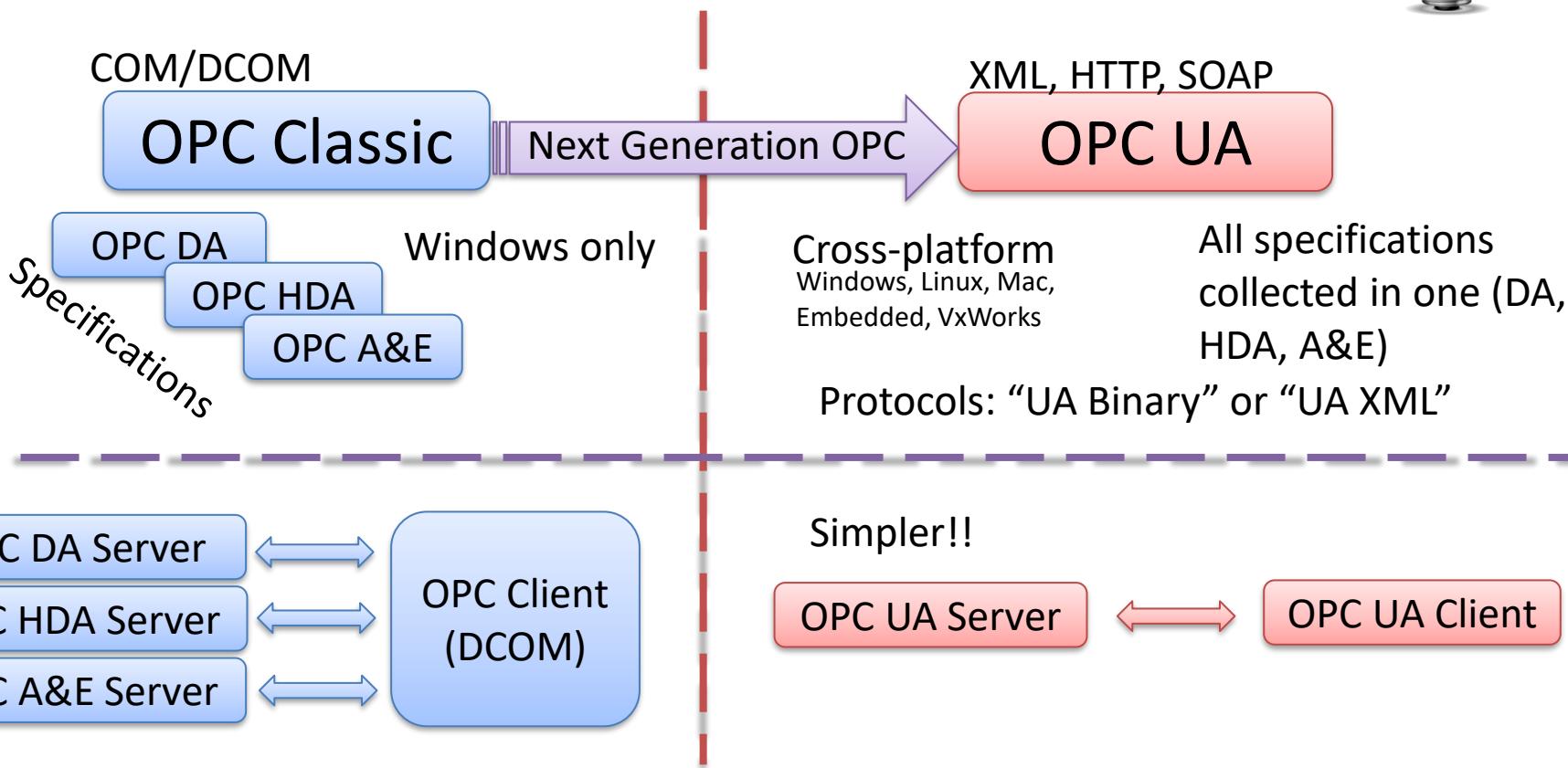
“Classic” OPC

“Next Generation” OPC



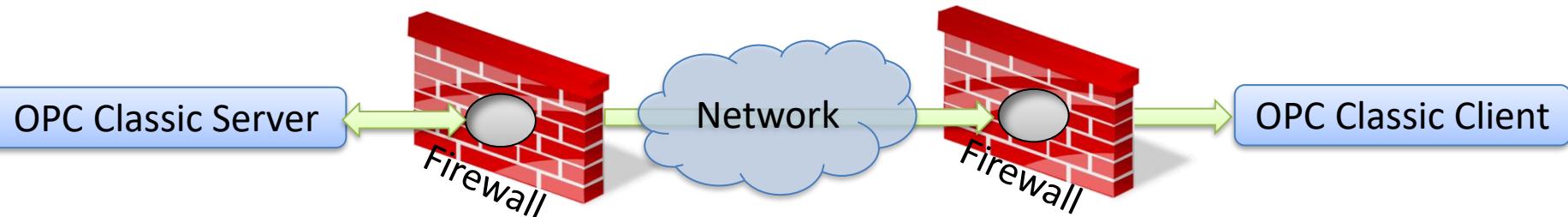
... (Many others)

Next Generation OPC

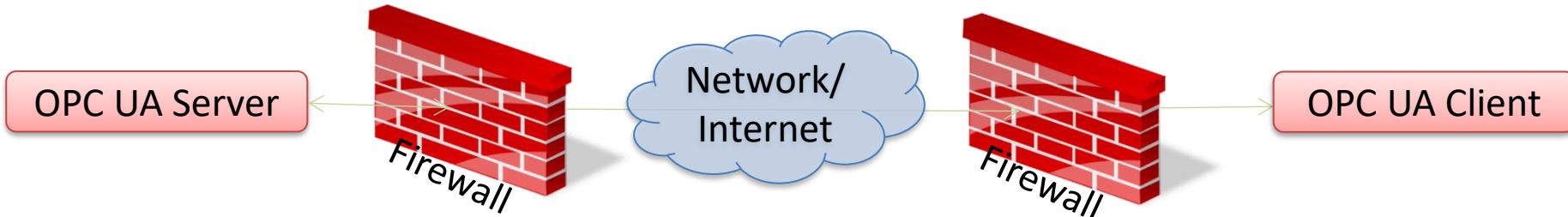




Next Generation OPC



To open DCOM through firewalls demanded a large hole in the firewall!
Impossible to route over Internet!



No hole in firewall (UA XML) or just a simple needle stick (UA Binary) is necessary
Easy to route over Internet!

<https://www.halvorsen.blog>



OPC UA Server Simulator

Hans-Petter Halvorsen

[Table of Contents](#)

OPC UA Server Simulator

- This free OPC UA Server tool supports data access and historical access information models of OPC UA.
- Consequently, it provides simulated real-time and historical data.
- It is possible to configure your own tags and the data simulation via CSV files.
- OPC UA clients can monitor real-time data and explore history data from this simulator.
- <https://opcfoundation.org/products/view/opc-ua-server-simulator>

OPC UA Server Simulator

The screenshot shows the OPC Foundation website. At the top, there is a banner for "OPC DAY INTERNATIONAL APR 25-29, 2022 DIGITAL EVENT" with a "REGISTER HERE" button. Below the banner is a navigation bar with links: About, Membership, Products, Certification, Markets & Collaboration, Resources, and News & Events. The main content area shows the "OPC UA Server Simulator" product page. It features a screenshot of the software interface, a member badge for Integration Objects, and a product website link. A descriptive text explains the simulator's purpose: "Integration Objects' OPC UA Server Simulator is a free to use and distribute OPC Unified Architecture server utility. Indeed, you can use this OPC UA simulator to play the role of OPC UA servers and test your OPC UA Client applications." Below this, another text block states: "This free OPC UA Server tool supports data access and historical access information models of OPC UA. Consequently, it provides simulated real-time and historical data. Moreover, users can configure their own tags and the data simulation via CSV files. OPC UA clients can monitor real-time data and explore history data from this simulator." At the bottom left is a "Back" button. On the right side, there are two boxes: "SUBSCRIBE NEWSLETTER" and "BECOME A MEMBER". Below these are two more boxes: "Newest Members" listing SAMSON, AKTIENGESELLSCHAFT Wuhan University, Transpara, CET Electric Technology Inc., and Linutronix GmbH; and "Certified Products" listing VMS OPCUA Server, ACCON-OPC-Server UA, PLCnext Controller AXC F 2152, and Collaborative Information Server.

<https://opcfoundation.org/products/view/opc-ua-server-simulator>

<https://opcfoundation.org>

[+ OPC Tunneling](#)[+ OPC UA](#)[▶ OPC UA Server Simulator –](#)[Full Edition](#)[▶ OPC UA Server Toolkit](#)[▶ OPC UA IoT Broker](#)[▶ OPC UA Server for
Databases](#)[▶ OPC UA Client Toolkit](#)[▶ OPC UA Server Simulator](#)[▶ OPC UA Proxy](#)[▶ OPC UA Wrapper](#)[▶ OPC UA Client](#)[+ OPC Data Archiving](#)[+ OPC Clients](#)[+ OPC Servers](#)[+ OPC Client Toolkits](#)[+ OPC Free Tools](#)[+ OPC Server Toolkits](#)

OPC UA Server Simulator

[Download](#)[User Guide](#)

[Watch Demo Videos](#)

Simulate real-time and historical data using OPC UA Server Simulator!

Integration Objects' **OPC UA Server Simulator** is a free to use and distribute OPC Unified Architecture server utility. Indeed, you can use this OPC UA simulator to play the role of OPC UA servers and test your OPC UA Client applications.

This free OPC UA Server tool supports data access and historical access information models of OPC UA. Consequently, it provides simulated real-time and historical data. Moreover, users can configure their own tags and the data simulation via CSV files. OPC UA clients can monitor real-time data and explore history data from this simulator.



OPC UA Server Simulator

[Privacy & Cookies Policy](#)

OPC UA Server Simulator

The screenshot shows the OPC UA Server Simulator application window. The title bar reads "OPC UA Server Simulator". The menu bar includes "File", "Settings", and "Help". A status bar at the bottom displays "Status: Running", "Current Time: 11:01:11", "Sessions: 0", "Subscriptions: 0", and "Items: 0".
The main content area contains two tables:

- Sessions**: A table with columns "SessionId", "Name", "User", and "Last Contact". It currently has no data.
- Subscriptions**: A table with columns "SubscriptionId", "Publishing Interval", "Item Count", and "Seq No". It currently has no data.

The URL "opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator" is displayed in the top right corner of the main window area.

OPC UA Server Simulator

The OPC UA Server Simulator uses 2 CSV simulation files:

- “**AddressSpace.csv**” used to build the address space of the OPC UA Server.
- “**ValueSpace.csv**” used to simulate the data values of the OPC UA items.
- Those two files are located at the following path:
X:\Program Files (x86)\Integration Objects\Integration Objects' OPC UA Server Simulator\OPC UA Server Simulator\DATA

AddressSpace.csv

A1	Tag Name
1	Tag Name
2	Tag1
3	Tag2
4	Tag3
5	Tag4
6	Tag5
7	Tag6
8	Tag7
9	Tag8
10	Tag9
11	Tag10
12	Tag11
13	Tag12
14	Tag13
15	Tag14
16	Tag15
17	Tag16
18	Tag17
19	Tag18
20	Tag19
21	Tag20
22	

ValueSpace.csv

A1	Tag11	Tag12	Tag13
1	11 good	56 good	47 good
2	12 good	32 good	14 good
3	13 good	28 good	85 good
4	14 good	14 good	125 good
5	15 good	15 good	24 good
6	16 good	57 good	69 good
7	17 good	65 good	36 good
8	18 good	18 good	18 good
9	19 good	48 good	84 good
10	20 good	36 good	64 good
11			
12			
13			
14			

AddressSpace

Ready	Accessibility: Unavailable
AddressSpace	

ValueSpace

Ready	Accessibility: Unavailable
ValueSpace	

<https://www.halvorsen.blog>



“OPC UA Client” Tool

Hans-Petter Halvorsen

[Table of Contents](#)

“OPC UA Client” Tool

- “OPC UA Client” is a free client tool that supports the main OPC Unified Architecture information models.
- These models are Data Access, Alarms & Conditions, and Historical Data Access
- <https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-client/>

[!\[\]\(f9185a5d5711fafb6778a681a6531e28_img.jpg\) OPC Tunneling](#)[!\[\]\(dbd160ae54b5a1956d3a6f28868e6bae_img.jpg\) OPC UA](#)

▶ OPC UA Server Simulator –
Full Edition

▶ OPC UA Server Toolkit

▶ OPC UA IoT Broker

▶ OPC UA Server for
Databases

▶ OPC UA Client Toolkit

▶ OPC UA Server Simulator

▶ OPC UA Proxy

▶ OPC UA Wrapper

[!\[\]\(b54da9a49273d0d34f6202d50bdc2544_img.jpg\) OPC UA Client](#)

▶ OPC Data Archiving

▶ OPC Clients

▶ OPC Servers

▶ OPC Client Toolkits

▶ OPC Free Tools

▶ OPC Server Toolkits

OPC UA Client

[Download](#)[User Guide](#)[Quick User Guide](#)

Download free OPC UA Client and start your OPC UA tests now!

OPC UA Client is a free client tool that supports the main OPC Unified Architecture information models. These models are Data Access, Alarms & Conditions, and Historical Data Access. In fact, it offers the capability to:

- ▶ Discover local and remote OPC UA servers
- ▶ Establish secure communication channels
- ▶ Browse the address space of any OPC UA compliant server
- ▶ Monitor real-time data and alarms & conditions
- ▶ Explore and update history data



Moreover, this OPC UA explorer allows you to generate its self-signed Application Instance Certificate in order to provide application level security and secure the connections with OPC UA servers.

▶ [View Tutorial Video of OPC UA Test Client & OPC UA Wrapper](#)



OPC UA Client



Home



Sessions

Session Name
Session0

Address Space

Forward

Path
/

Connection Settings

Session Information

Session Name: Session0

Server Information

Endpoint Url: opc.tcp://xps15hph:62640/IntegrationObjects/ [Discover]

Transport Protocol

Opc.tcp
Https

Message Encoding

Binary
Xml

Security Mode

None
Sign
Sign_Encrypt

Security Policy

None
Basic128RSA15
Basic256
Basic256Sha256

User Authentication Mode

Anonymous
UserName
Certificate

Certificate (.pfx):

Password:

Buttons: Apply, Cancel

Attribute Value

Attribute	Value
Session	Session0
Subscription	
Session	

Message Log:

[Control] 2022-02-08 13:05:06 Disconnecting from session 'Session0'

[Control] 2022-02-08 13:03:09 Read operation of the variable 'Session' was successful.

[Control] 2022-02-08 13:01:03 A session "Session0" with the URL "opc.tcp://xps15hph:62640/IntegrationObjects/" was successfully created.

3 Messages

Home

New
Open
Save
Save as
Connect
Disconnect
Settings
UA Settings
Help
About
Define
Remove
Certificate Manager

Sessions

- Session0

Data View **History View** **Event View**

Display Name	Node Id	Value	Data Type	Server Timestamp	Source Timestamp	Status Code	Subscription	Session

Attribute	Value
Nodeld	ns=2;s=Historical...
NodeClass	Object
BrowseName	2:Historicaldata
DisplayName	Historical Data
Description	
WriteMask	0
UserWriteMask	0
EventNotifier	Subscribe

Address Space

Forward

- Real Time Data
 - Tag1
 - Tag2
 - Tag3
 - Tag4
 - Tag5
 - Tag6
 - Tag7
 - Tag8
 - Tag9

Read

- References and Attributes
- Read
- Write
- History Update
- Monitor

Message Type

Message Type	Timestamp	Message
[Control]	2022-02-08 13:03:09	Read operation of the variable [ns=2;s=Tag7] succeeded.
[Control]	2022-02-08 13:01:03	A session "Session0" with the Endpoint [opc.tcp://xps15hph:62640/IntegrationObjects/ServerSimulator - [None,None:Binary]] was successfully created.

2 Messages

<https://www.halvorsen.blog>



OPC UA with C#

Hans-Petter Halvorsen

[Table of Contents](#)

OPC UA with Visual Studio/C#

- Lots of Packages and Libraries do exist for creating both OPC UA Clients and OPC UA Servers in Visual Studio/C#
- Most of them are payment based
- Many of those can be evaluated for a trial period or used forever with some restrictions
- In this Tutorial, “OPC UA .NET SDK” will be used

<https://www.halvorsen.blog>



OPC UA .NET SDK

Hans-Petter Halvorsen

[Table of Contents](#)

OPC UA .NET SDK

- The “OPC UA .NET SDK” comes with an evaluation license which can be used unlimited for each application run for 30 minutes
- It comes in a NuGet Package you can install and use in your Visual Studio Project
- <https://opcfoundation.org/products/view/opc-ua-net-sdk-for-client-and-server>

Products » OPC UA .NET SDK for Client and Server

OPC UA .NET SDK for Client and Server



Member: Traeger Industry Components GmbH

Product website: opcua.traeger.de

OPC UA
Client & Server
in C# / VB.NET
quick and easy.

Introduction: <https://opcua.traeger.de/>

Development: <https://docs.traeger.de/en/software/sdk/opc-ua/net/>

NuGet Package: <https://www.nuget.org/packages/Opc.UaFx.Advanced/>

Samples: <https://github.com/Traeger-GmbH/opcua.net-samples/>

Description

The OPC UA .NET SDK allows rapid and easy development of Client and / or Server applications using .NET. With a few lines of code you can realize your application in minutes. The SDK is provided for .NET Standard 2.0+, .NET Core 3+ and .NET Framework 4.6+. Therefore the SDK supports Windows, Linux, macOS, Android, iOS and Unity. No installation required, just download the ZIP or NuGet package and get started.

Features

- OPC UA with DA, AE, HDA and more
- OPC UA Companion Specifications
- OPC Classic (with just a different URI)

<https://www.halvorsen.blog>



Visual Studio/C# Example

Hans-Petter Halvorsen

[Table of Contents](#)

Visual Studio/C# Example

The screenshot shows a Windows application window titled "OPC UA Client". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there are two sections. The top section is labeled "Enter Value:" and contains a text input field with the value "22.4" and a button labeled "Write OPC". The bottom section is labeled "OPC Value:" and contains a text input field with the value "22.4" and a button labeled "Read OPC". The "Read OPC" button is highlighted with a blue border.

OPC UA Client

Enter Value:

22.4

Write OPC

OPC Value:

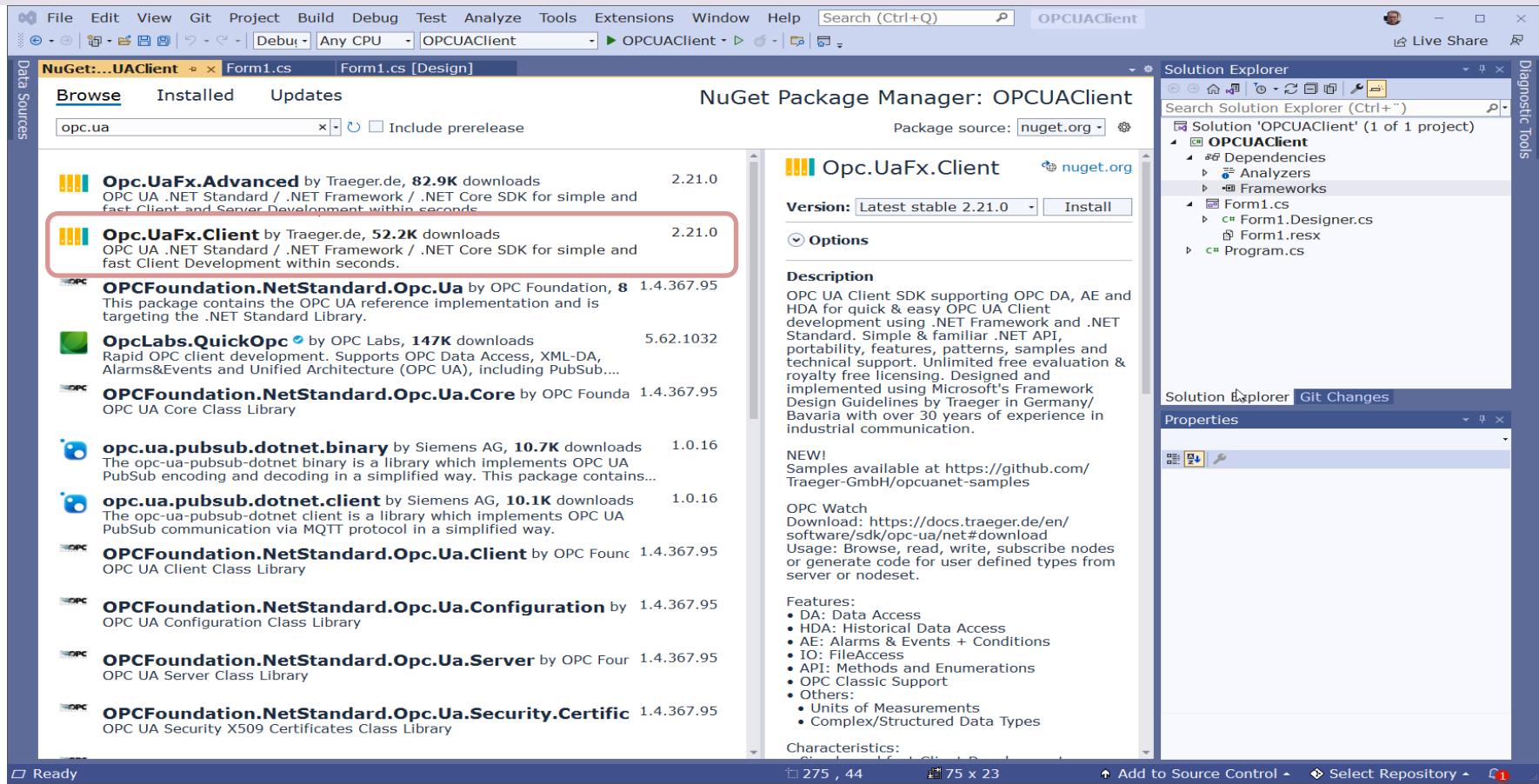
22.4

Read OPC

Visual Studio/C# Example

- Note that this is a simplified example in order demonstrate the principle of Writing Data to an OPC UA Server and Reading Data from an OPC Server
- Write and Read the same OPC Tag in the same Application makes no sense in a real scenario
- Typically, the OPC clients are distributed in a network and the different Applications are located on different computers in a network

NuGet Package



Visual Studio Project

The screenshot shows the Visual Studio IDE interface with the following components:

- Top Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, Search (Ctrl+Q).
- Solution Explorer:** Shows the solution 'OPCUAClient' with one project 'OPCUAClient' containing files: Dependencies, Analyzers, Frameworks, Packages, Form1.cs, Form1.Designer.cs, Form1.resx, and Program.cs.
- Properties:** Shows icons for file, folder, and key.
- Code Editor:** The main window displays the 'Form1.cs [Design]' file. The code uses the `Opc.UaFx.Client` namespace to interact with an OPC UA server. It includes methods for writing data to a node and reading data from a node. A red box highlights the `using Opc.UaFx.Client;` statement at the top.
- Error List:** Shows 0 Errors, 0 Warnings, and 0 of 2 Messages.
- Output:** Shows the build output.

OPC UA Write

```
private void btnOpcWrite_Click(object sender, EventArgs e)
{
    string opcUrl = "opc.tcp://localhost:62640/";
    var tagName = "ns=2;s=Tag7";

    var client = new OpcClient(opcUrl);
    client.Connect();

    double temperature;
    temperature = Convert.ToDouble(txtOpcDataWrite.Text);

    client.WriteNode(tagName, temperature);

    client.Disconnect();
}
```

OPC UA Read

```
private void btnOpcRead_Click(object sender, EventArgs e)
{
    string opcUrl = "opc.tcp://localhost:62640/";
    var tagName = "ns=2;s=Tag7";

    var client = new OpcClient(opcUrl);
    client.Connect();

    var temperature = client.ReadNode(tagName);
    txtOpcDataRead.Text = temperature.ToString();

    client.Disconnect();
}
```

Test 1 (OPC UA Server Simulator)

The diagram illustrates the communication flow between an OPC UA Server Simulator and an OPC UA Client.

OPC UA Server Simulator:

- Shows the "Sessions" and "Subscriptions" tabs.
- The "Sessions" tab lists "Session0" (Anonymous) with details: ns=3;i=1603066462, Last Contact: 11:21:07.
- The "Subscriptions" tab shows a single subscription with SubscriptionId 1, Publishing Interval 1000, Item Count 1, and Seq No 3.
- Status: Running

OPC UA Client:

- Shows the "Enter Value:" field containing "22".
- A "Write OPC" button is present.
- Shows the "OPC Value:" field containing "22".
- A "Read OPC" button is present.

Integration Objects' OPC UA Client:

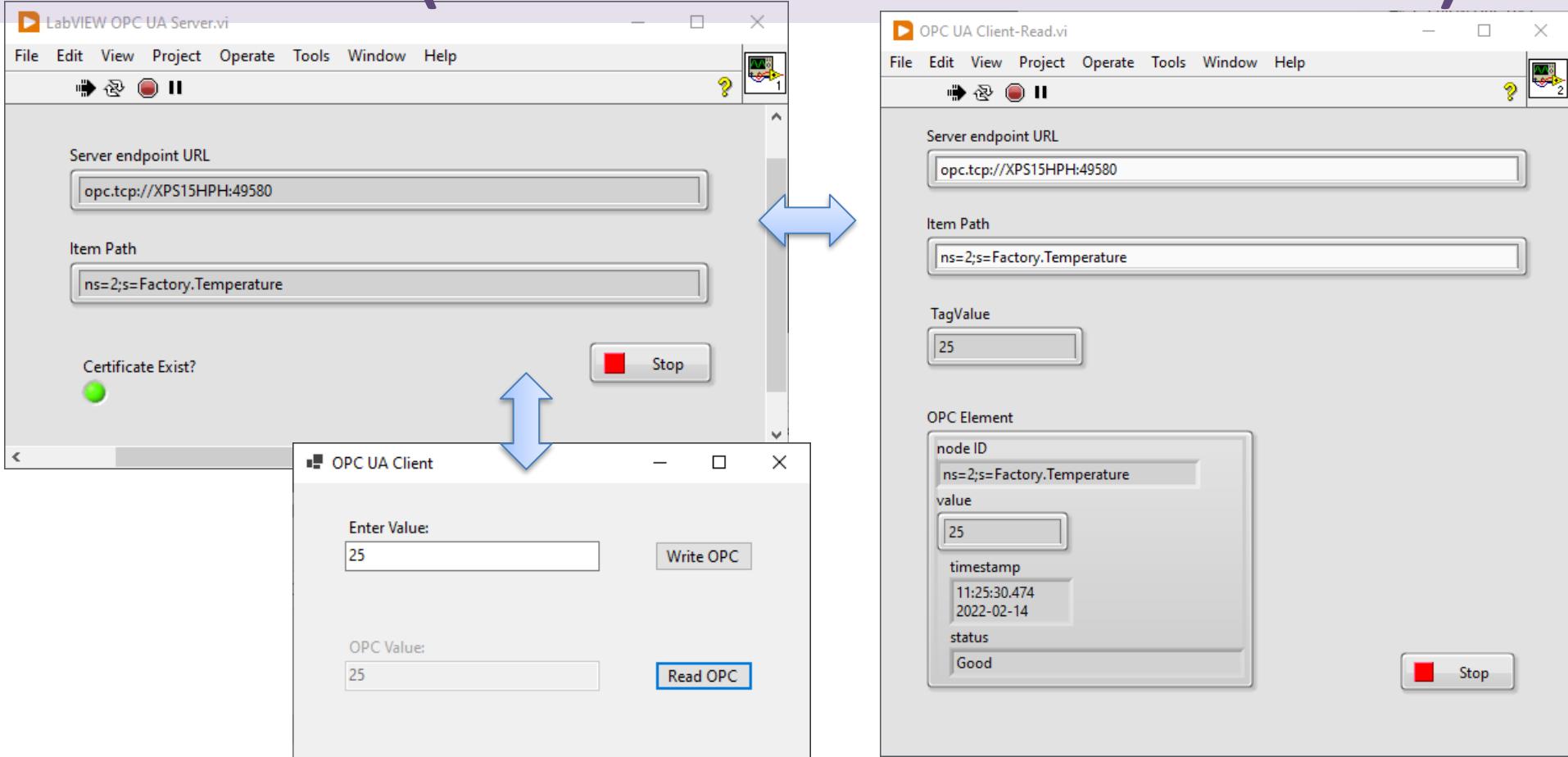
- Shows the "Sessions" and "Data View" tabs.
- The "Sessions" tab shows "Session0" and "Subscription0".
- The "Data View" tab displays a table of data:

Display Name	Node Id	Value	Data Type	Server Timestamp	Source Timestamp	Status Code	Subscription	Session
Tag7	ns=2#Tag7	22	Double	14-02-2022 1...	14-02-2...	Good	Subscription0	Session0
- Address Space tree showing "Real Time Data" with nodes Tag1, Tag2, Tag3, Tag4, Tag5, Tag6, Tag7, and Tag8.

Communication Flow:

- A double-headed arrow indicates bidirectional communication between the OPC UA Server Simulator and the OPC UA Client.
- A large blue double-headed arrow indicates bidirectional communication between the OPC UA Server Simulator and the Integration Objects' OPC UA Client.

Test 2 (LabVIEW OPC Server)



<https://www.halvorsen.blog>



Improved Example

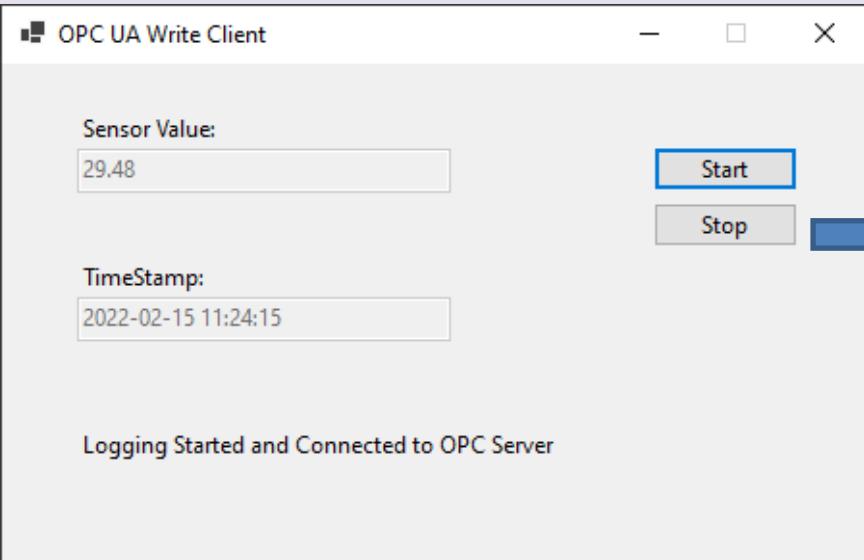
Hans-Petter Halvorsen

[Table of Contents](#)

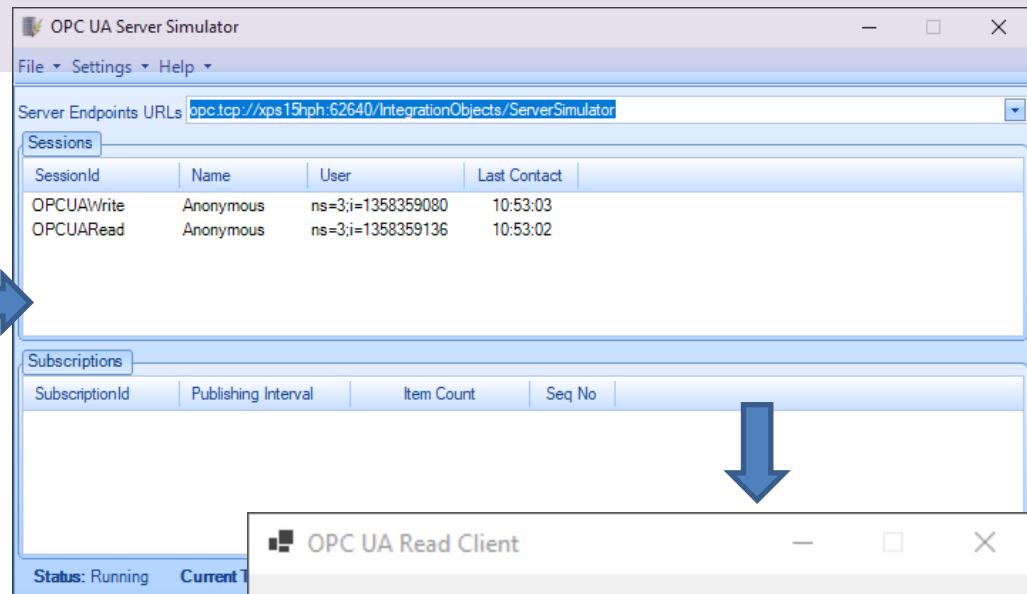
Improved Example

We will implement some Improvements:

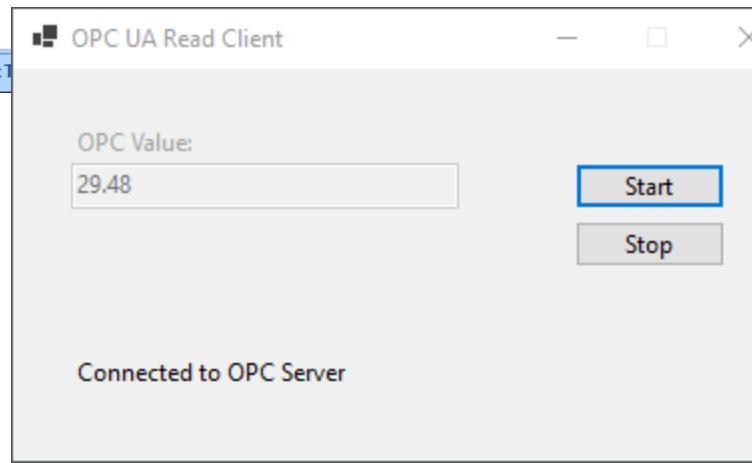
- Separate Applications for Write and Read
- Start and Stop Buttons
- Formatting Number of Decimals
- Property window (preventing resizing of the window):
 - `form1.FormBorderStyle = FormBorderStyle.FixedSingle;`
 - `Form1.MaximizeBox = false;`
- Using a Timer (Writing/Reading at specific Intervals)
- General Improvements in the Code



OPC UA Write C# App



OPC UA Read C# App



OPC UA Write

```
using Opc.UaFx.Client;
namespace OPCUARead
{
    public partial class Form1 : Form
    {
        public OpcClient client = new OpcClient("opc.tcp://localhost:62640/");

        public Form1()
        {
            InitializeComponent();
            timer1.Interval = 10000;
        }

        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (client != null)
                client.Disconnect();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            double sensorValue;

            sensorValue = ReadSensorData();
            OpcWrite(sensorValue);
        }

        double ReadSensorData()
        {
            var rand = new Random();
            int minValue = 20, maxValue = 30;
            double sensorValue;

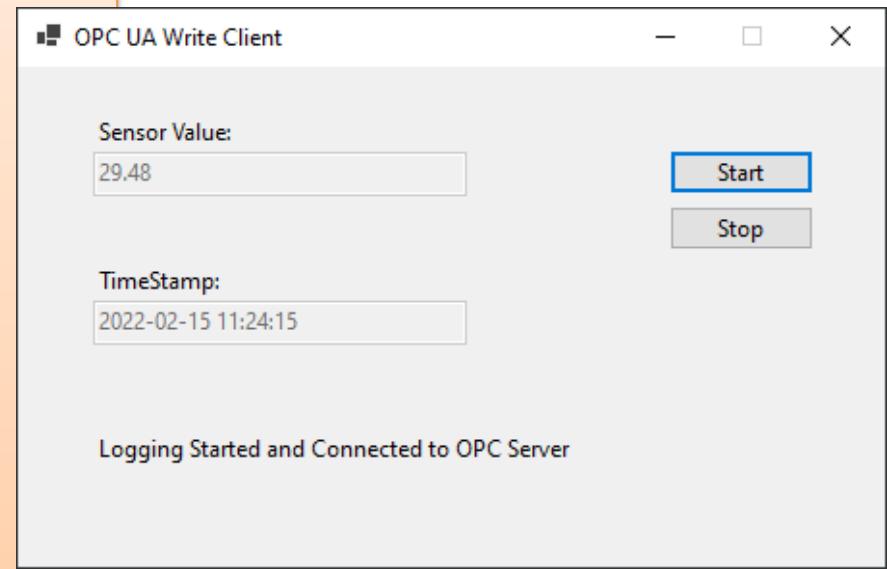
            sensorValue = rand.NextDouble() * (maxValue - minValue) + minValue;
            txtSensorValue.Text = sensorValue.ToString("#.##");
            DateTime sensorDateTime = DateTime.Now;
            txtTimeStamp.Text = sensorDateTime.ToString("yyyy-MM-dd HH:mm:ss");

            return sensorValue;
        }

        void OpcWrite(double sensorValue)
        {
            string tagName = "ns=2;s=Tag";
            client.WriteNode(tagName, sensorValue);
        }

        private void btnStart_Click(object sender, EventArgs e)
        {
            client.Connect();
            timer1.Start();
            lblStatusMessage.Text = "Logging Started and Connected to OPC Server";
        }

        private void btnStop_Click(object sender, EventArgs e)
        {
            timer1.Stop();
            if (client != null)
                client.Disconnect();
            lblStatusMessage.Text = "Logging Stopped and Disconnected from OPC Server";
        }
    }
}
```



OPC UA Read

```
using Opc.UaFx.Client;

namespace OPCUARead
{
    public partial class Form1 : Form
    {
        public OpcClient client = new OpcClient("opc.tcp://localhost:62640/");

        public Form1()
        {
            InitializeComponent();
            timer1.Interval = 2000;
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            OpcRead();
        }

        void OpcRead()
        {
            string tagName = "ns=2;s=Tag7";

            Opc.UaFx.OpcValue opcData = client.ReadNode(tagName);

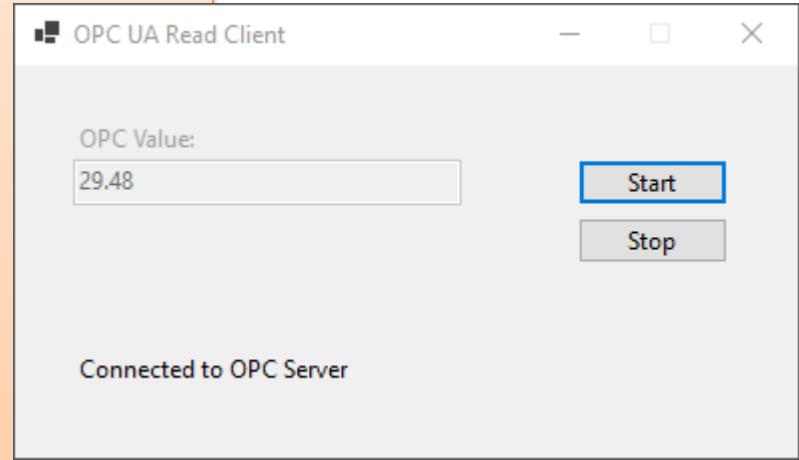
            double temperature = (double)opcData.Value;

            txtOpcValue.Text = temperature.ToString("#.###");
        }

        private void btnStart_Click(object sender, EventArgs e)
        {
            client.Connect();
            timer1.Start();
            lblStatusMessage.Text = "Connected to OPC Server";
        }

        private void btnStop_Click(object sender, EventArgs e)
        {
            timer1.Stop();
            if (client != null)
                client.Disconnect();
            lblStatusMessage.Text = "Disconnected from OPC Server";
        }

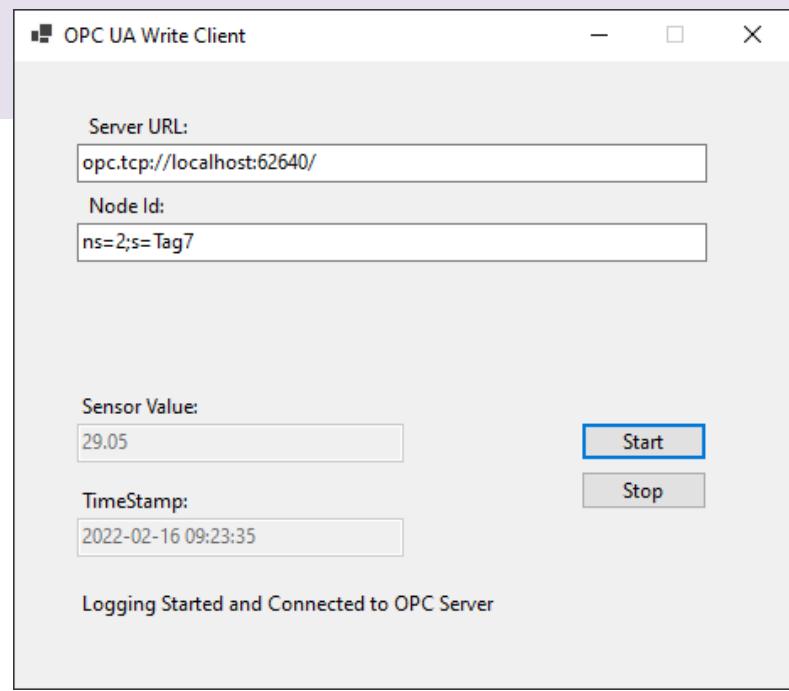
        private void Form1_FormClosing(object sender, FormClosingEventArgs e)
        {
            if (client != null)
                client.Disconnect();
        }
    }
}
```



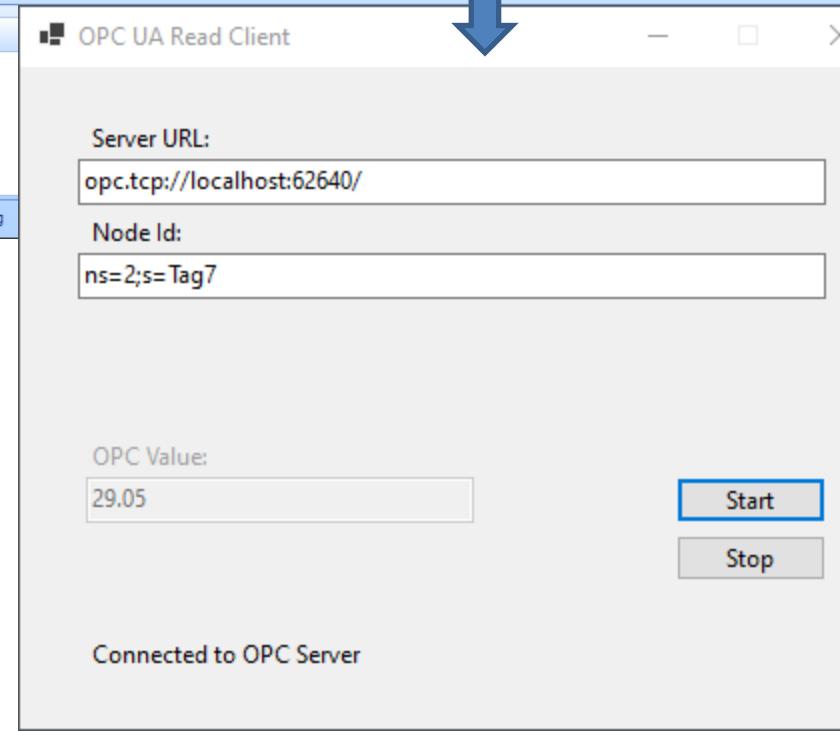
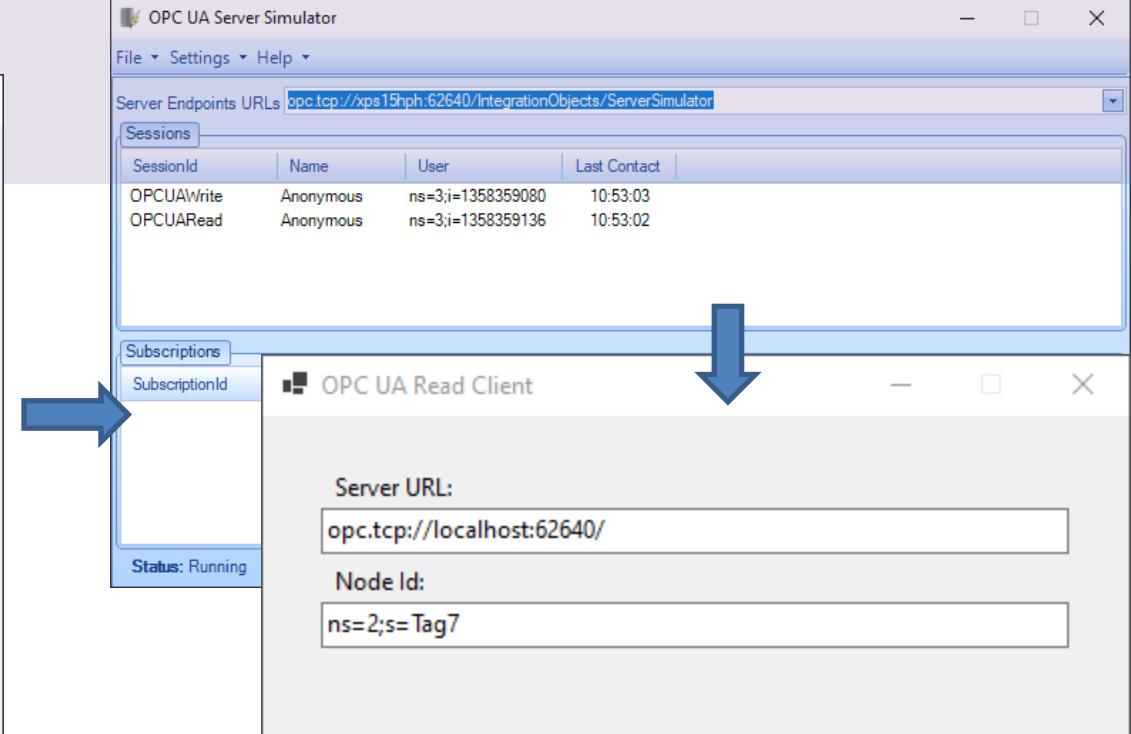
More Improvements

We will implement some more Improvements:

- Possible to specify ServerUrl and NodId
(TagName) from the GUI



OPC UA Write C# App



OPC UA Read C# App

Further Work

Still, lots of improvements can be made

- Possible to change the Logging Interval from the GUI
- Add a Chart
- Get Data from a real Sensor
- Add Units
- Etc.
- In addition, the “OPC UA .NET SDK” comes with many features that will not be demonstrated here

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

