https://www.halvorsen.blog

OPC

with Practical Examples

Hans-Petter Halvorsen

Contents

Introduction

- In this Tutorial we give an overview of OPC with some Practical Examples
- We use different OPC Software and different types of Programming Languages and Tools

What is OPC?

Hans-Petter Halvorsen

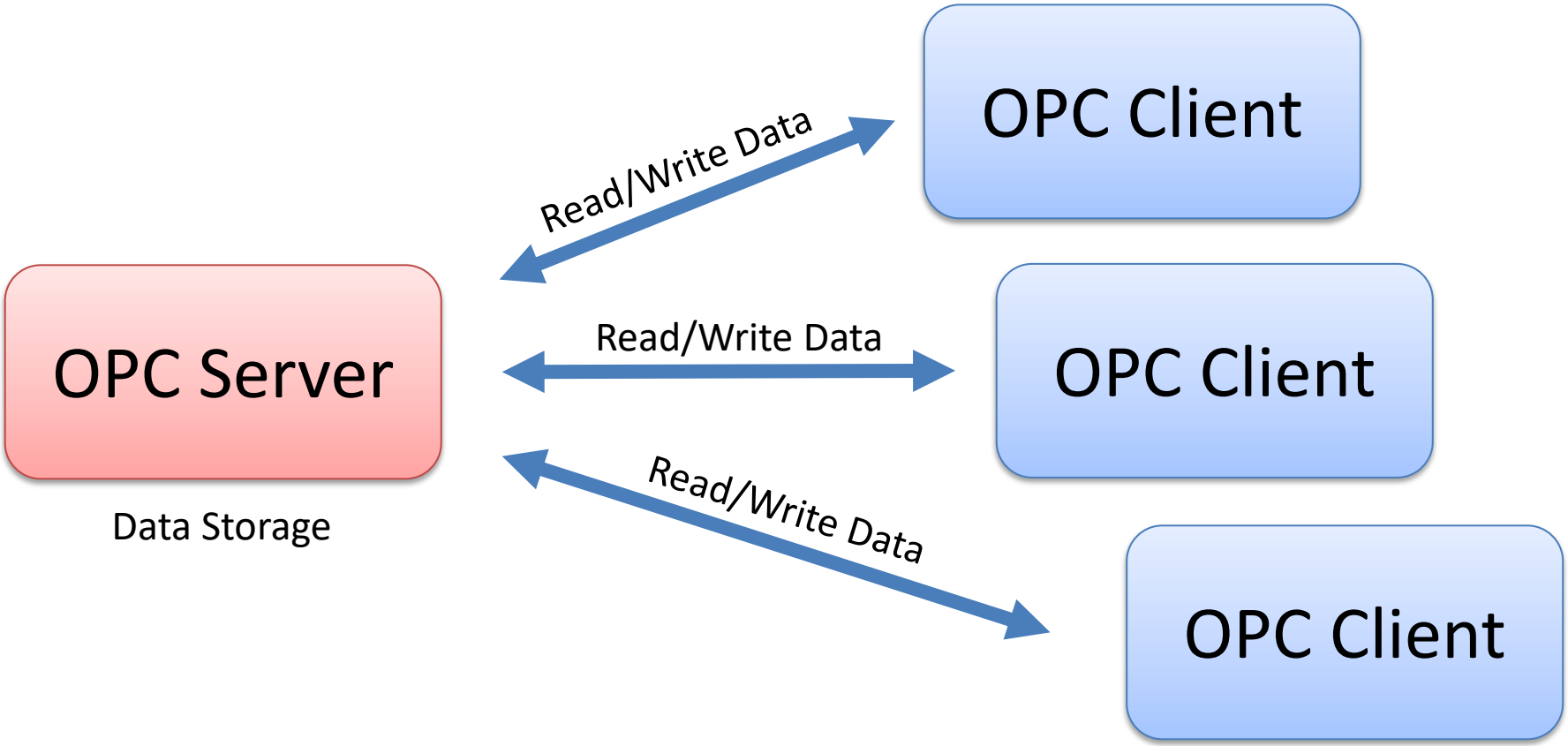
Table of Contents

What is OPC?

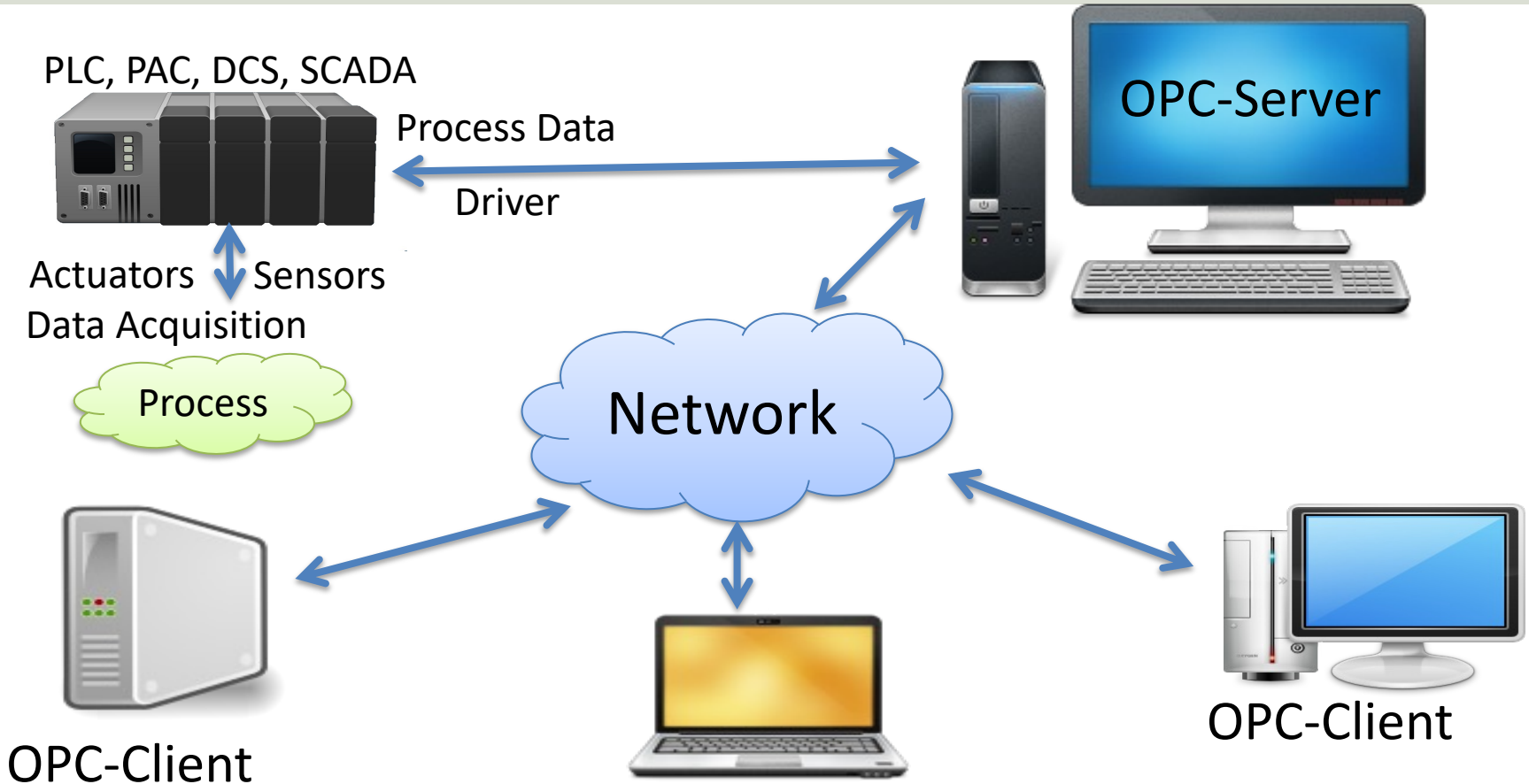
- OPC is a standard that defines the communication of data between devices from different manufactures
- OPC requires an **OPC Server** that communicates with one or more **OPC Clients**
- OPC allows “plug-and-play”, gives benefits as reduces installation time and the opportunity to choose products from different manufactures
- We have different OPC standards:
  - “Real-time” data (OPC DA),
  - Historical data (OPC HDA)
  - Alarm & Events data (OPC A&E)
  - etc.

https://opcfoundation.org/about/what-is-opc/

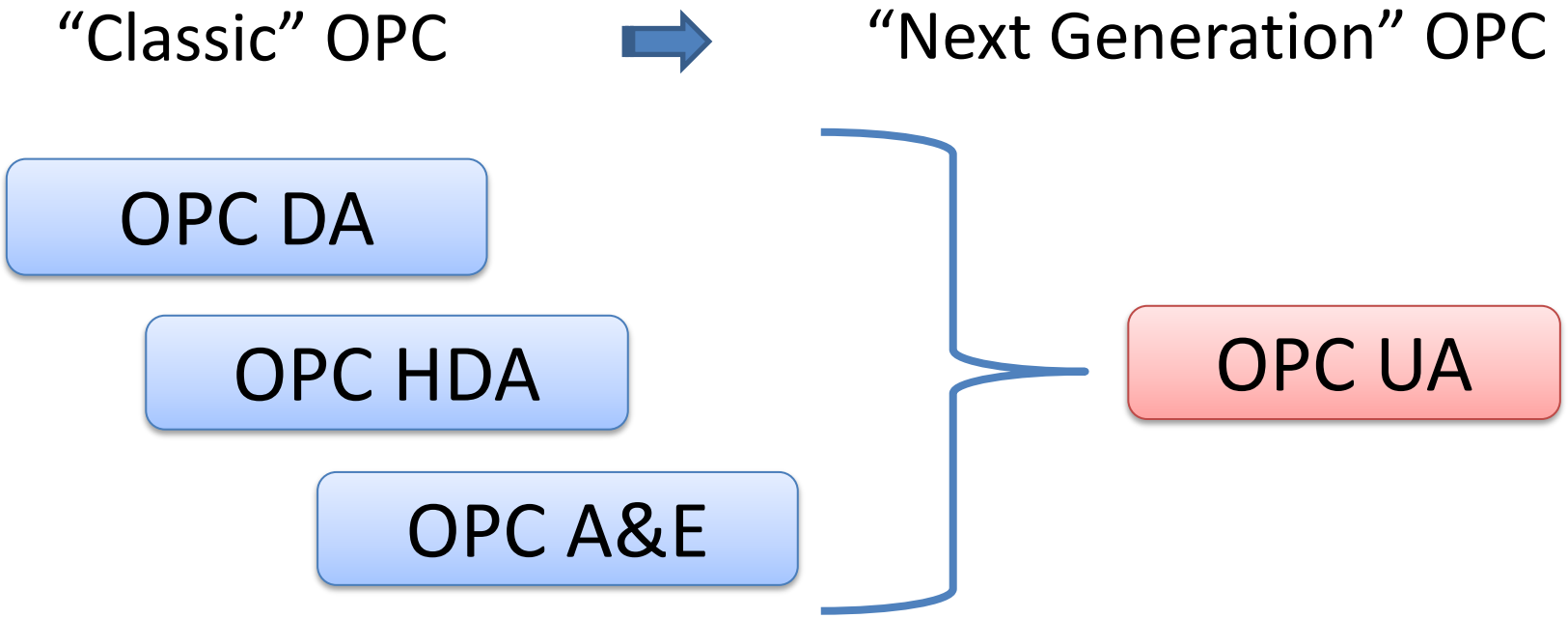
OPC Server and Clients



Typical OPC Scenario

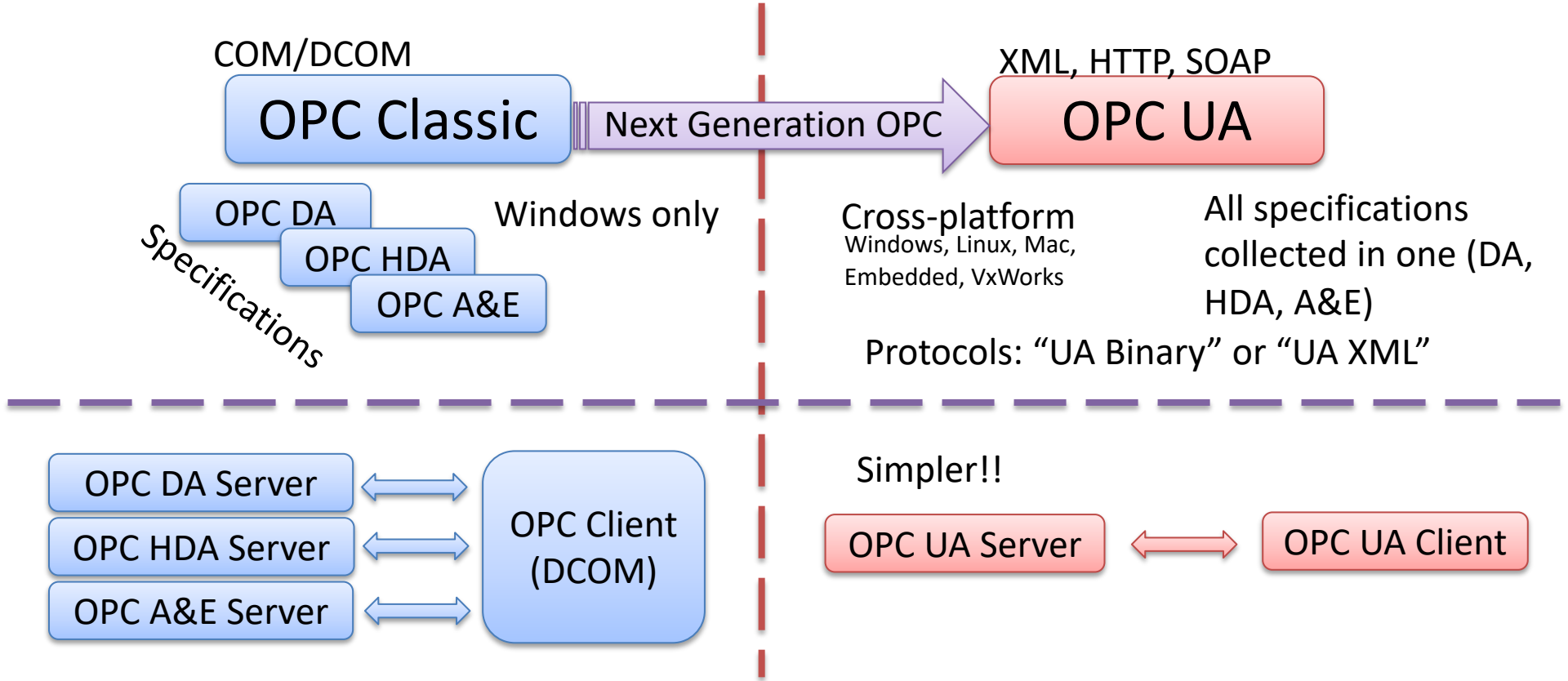


OPC Specifications



https://opcfoundation.org/about/opc-technologies/opc-classic/

Next Generation OPC



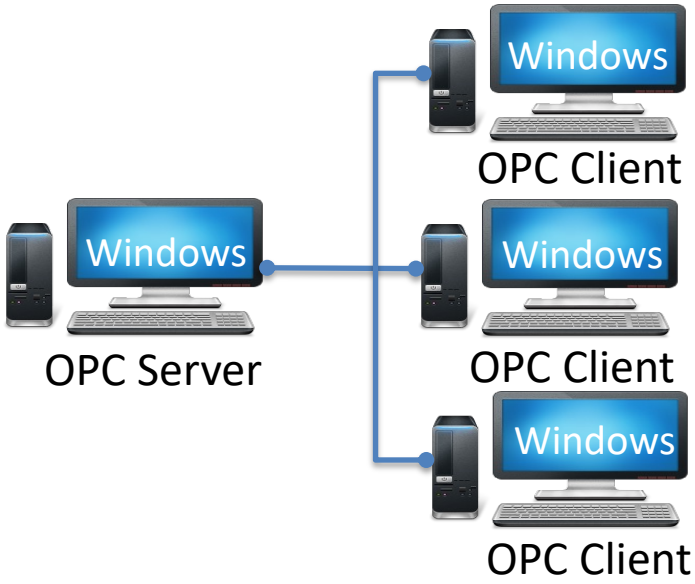
OPC UA

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

https://opcfoundation.org/about/opc-technologies/opc-ua/

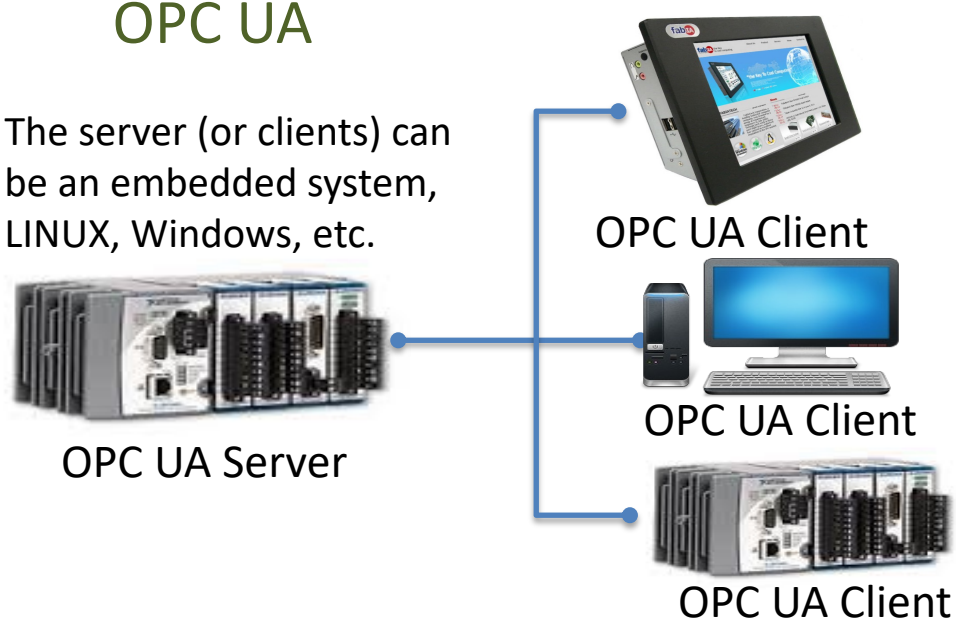
Classic OPC vs. OPC UA

Classic OPC (DCOM)



OPC UA

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



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 DA

Hans-Petter Halvorsen

Table of Contents

OPC DA Servers

Hans-Petter Halvorsen

Table of Contents

OPC DA Servers

- “MatrikonOPC Simulation Server” by Matrikon
- “OPC Server Simulators” by Integration Objects
- “NI OPC Servers” by NI

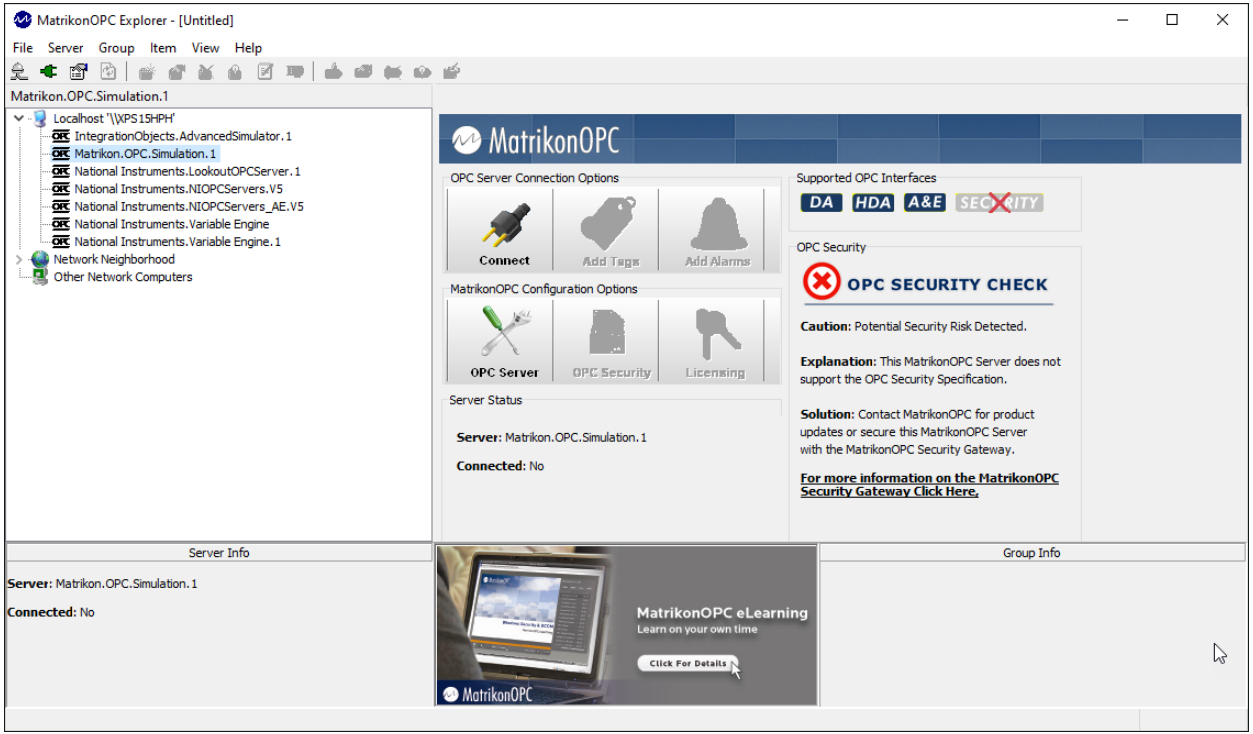
There exists hundreds of different OPC DA Servers, but those mentioned here can be used for free for demo and test purposes

MatrikonOPC Simulation Server

Hans-Petter Halvorsen

Table of Contents

MatrikonOPC Simulation Server



"MatrikonOPC Simulation Server" is free and can be used for testing and development of OPC DA solutions.

In the software package there is an OPC Test Client called "**MatrikonOPC Explorer**" which you can use to test communication with the OPC DA Server

https://www.matrikonopc.com/products/opc-drivers/opc-simulation-server.aspx

https://www.halvorsen.blog

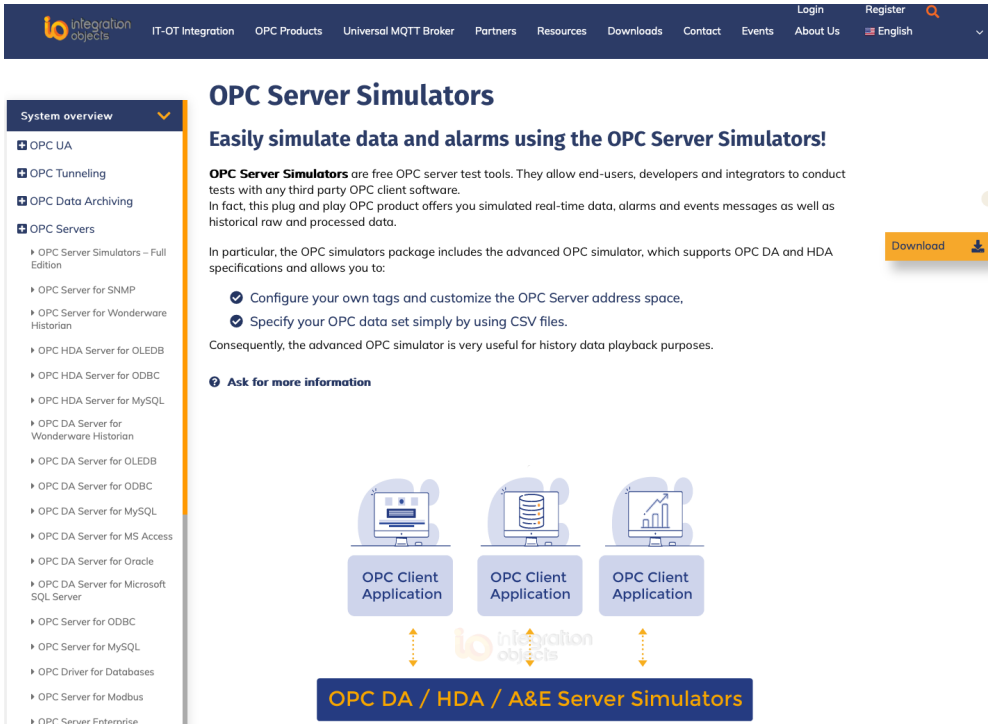
“OPC Server Simulators”

by Integration Objects

Hans-Petter Halvorsen

Table of Contents

“OPC Server Simulators”

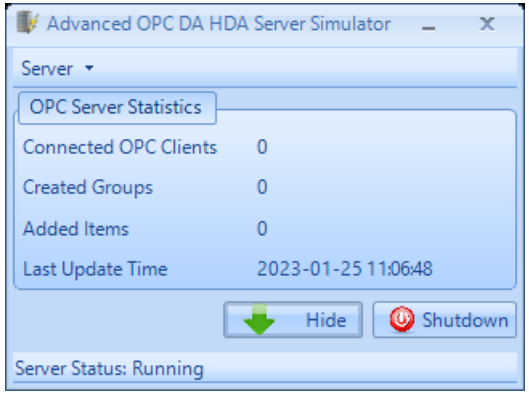


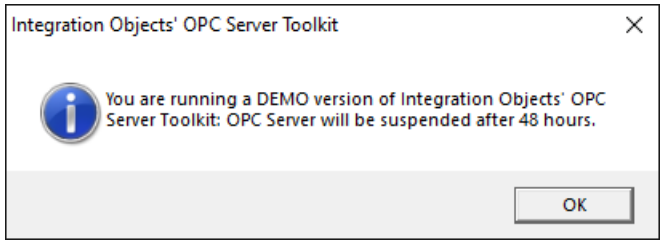
“OPC Server Simulators“ by Integration Objects are free OPC server test tools. They allow end-users, developers and integrators to conduct tests with any third party OPC client software.
The package consists of an OPC DA/HDA Server and an OPC A&E Server.

https://integrationobjects.com/sioth-opc/sioth-opc-servers/opc-server-simulators/

“OPC Server Simulators”

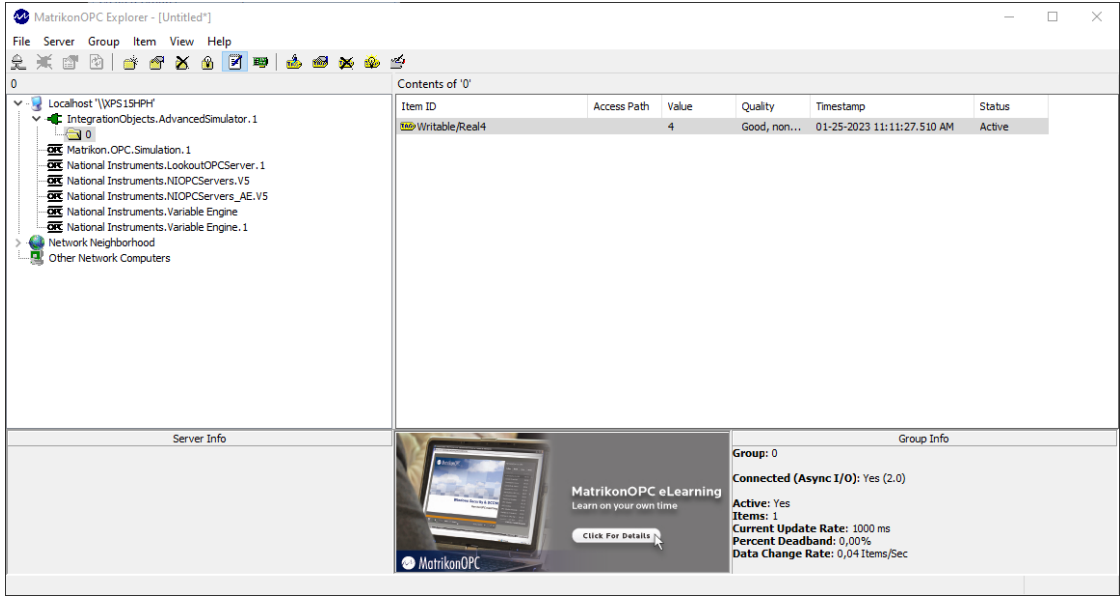
In the “OPC Server Simulators” software package there is a OPC server called “Advanced OPC DA HDA Server Simulator”





The demo version runs for 48 hours before you need to restart it

Here we have connected to the server using the “MatrikonOPC Explorer”:

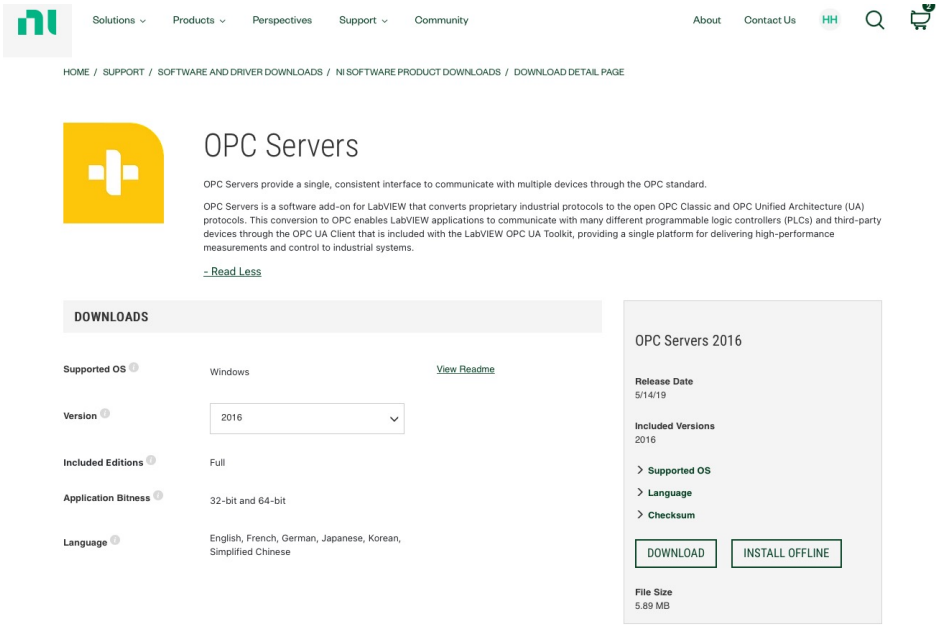


NI OPC Servers

Hans-Petter Halvorsen

Table of Contents

NI OPC Servers



A Demo version of “NI OPC Servers” is included with “LabVIEW DSC Module” or “LabVIEW Real-Time module” (so you may already have it installed on your PC).
It can also be downloaded separately.

https://www.ni.com/en-no/support/downloads/software-products/download.opc-servers.html

https://www.halvorsen.blog

OPC DA Programming Tools

Hans-Petter Halvorsen

Table of Contents

OPC DA Programming Tools

Software and Programming Tools that can be used for communicating with OPC DA Servers:

- LabVIEW + DataSocket
- MATLAB + Industrial Communication Toolbox (supports both OPC DA and UA)
- Visual Studio/C# + Measurement Studio

Many other alternatives exists

LabVIEW + DataSocket

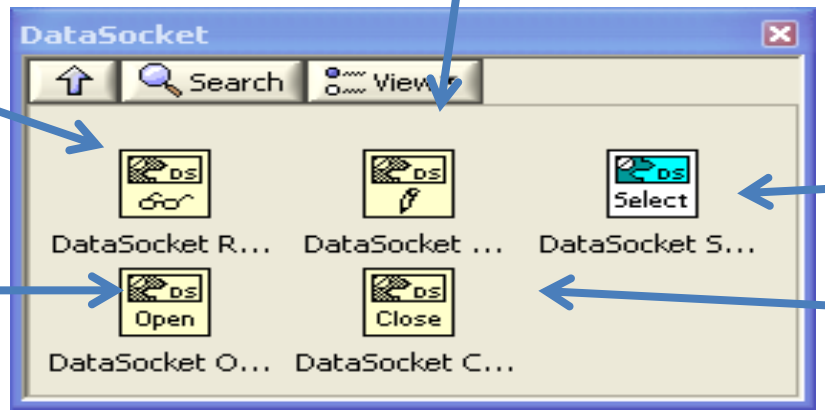
Hans-Petter Halvorsen

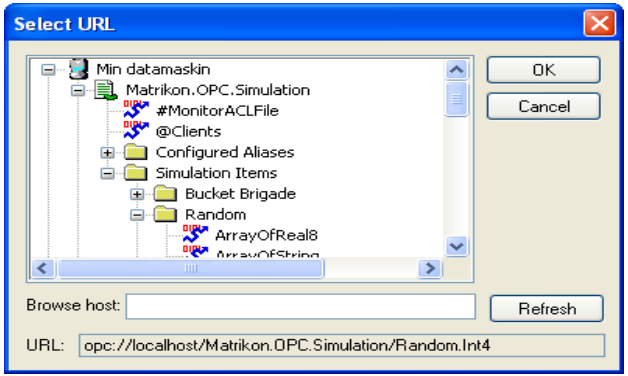
Table of Contents

OPC DA in LabVIEW

You can use LabVIEW as an OPC client by connecting to an OPC server through a DataSocket connection.

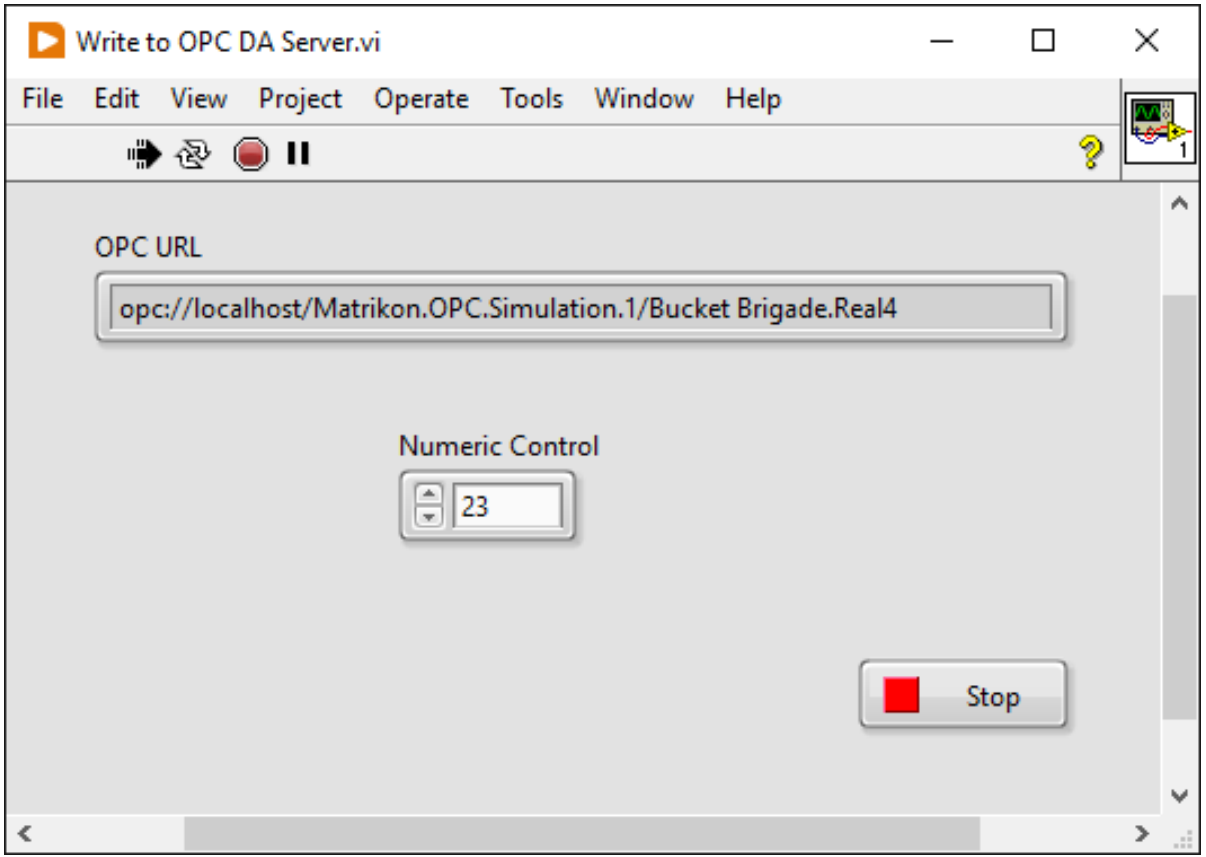
The **DataSocket** palette in LabVIEW:



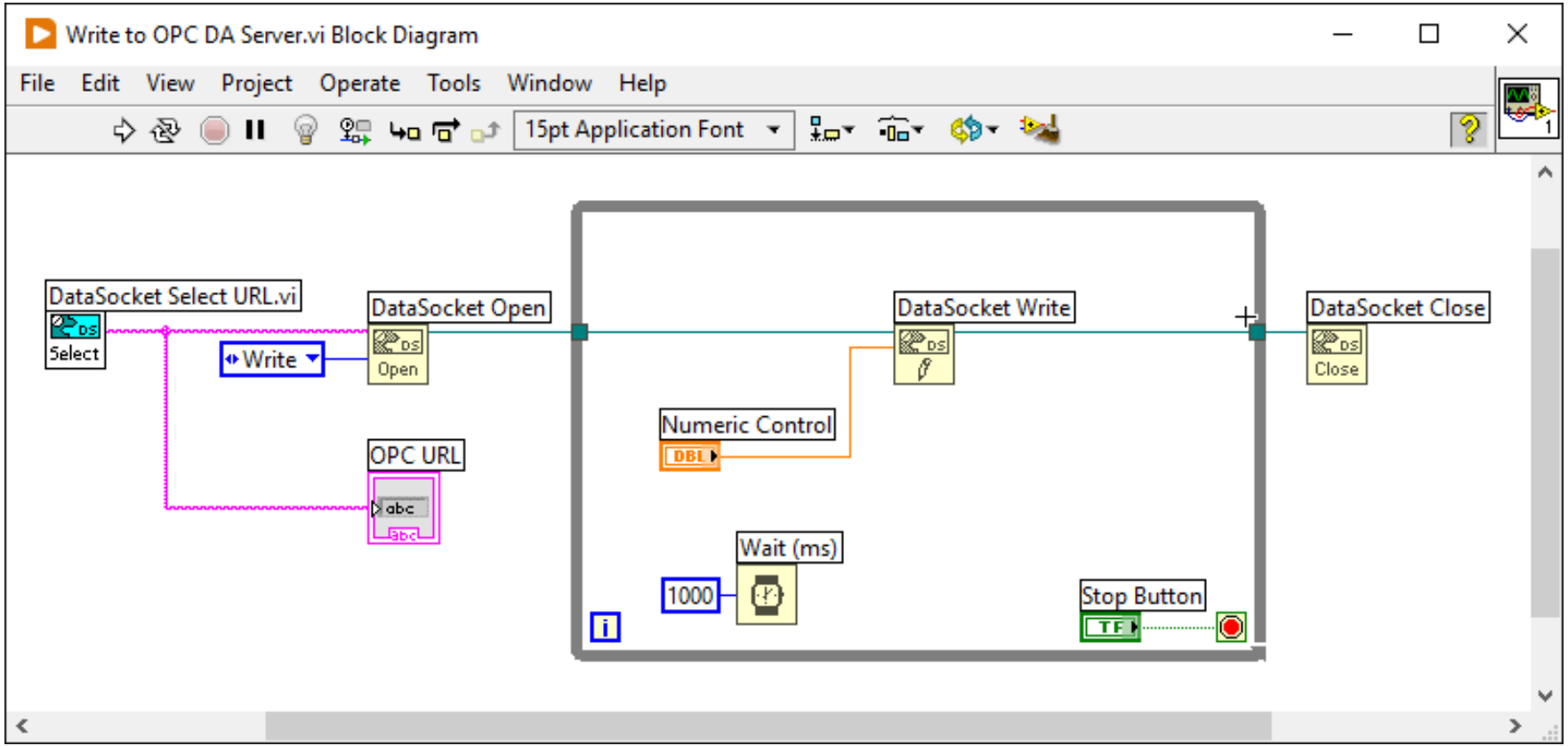


Note! Make sure to use **LabVIEW 32bit version** (even if you have 64bit operating system) because the DataSocket feature is only supported by the 32bit version of LabVIEW.

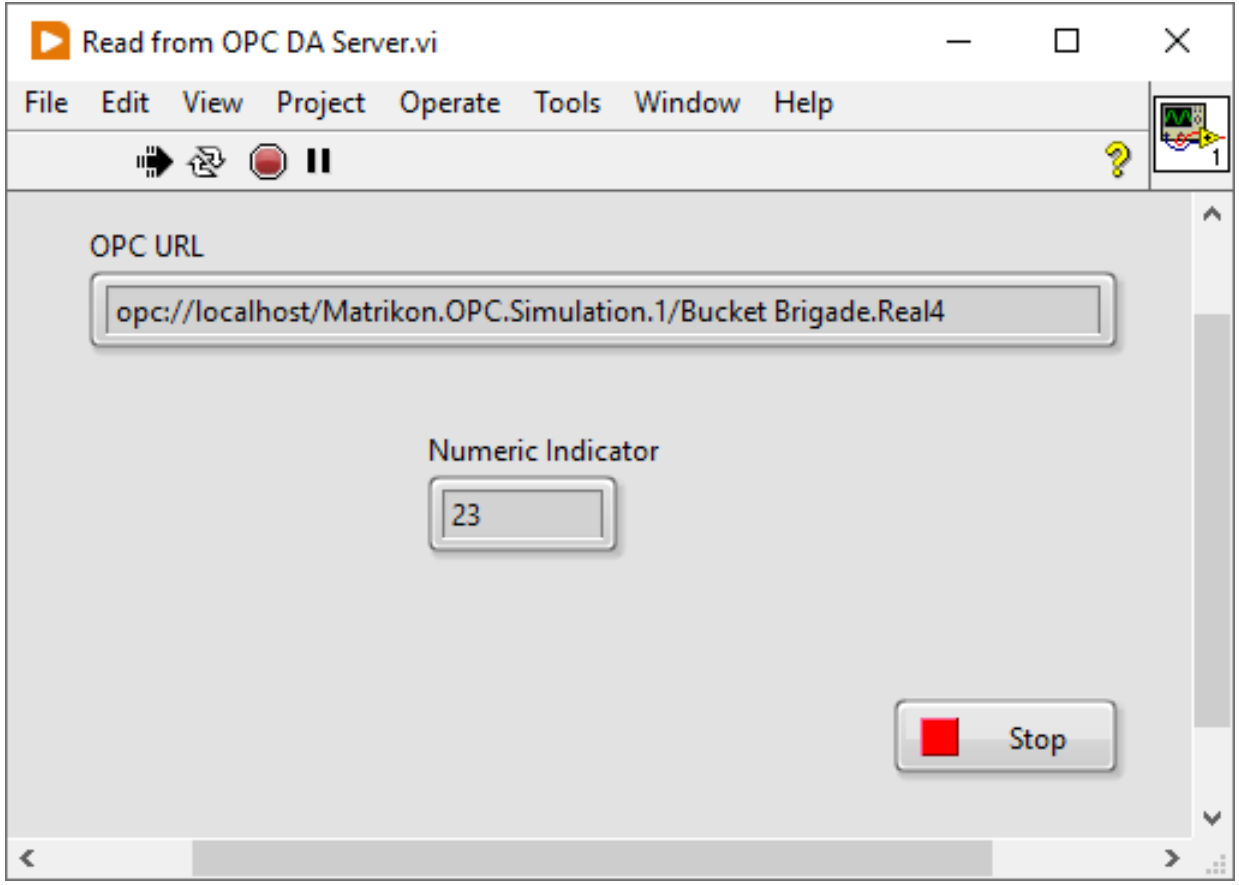
Write Data to OPC DA Server



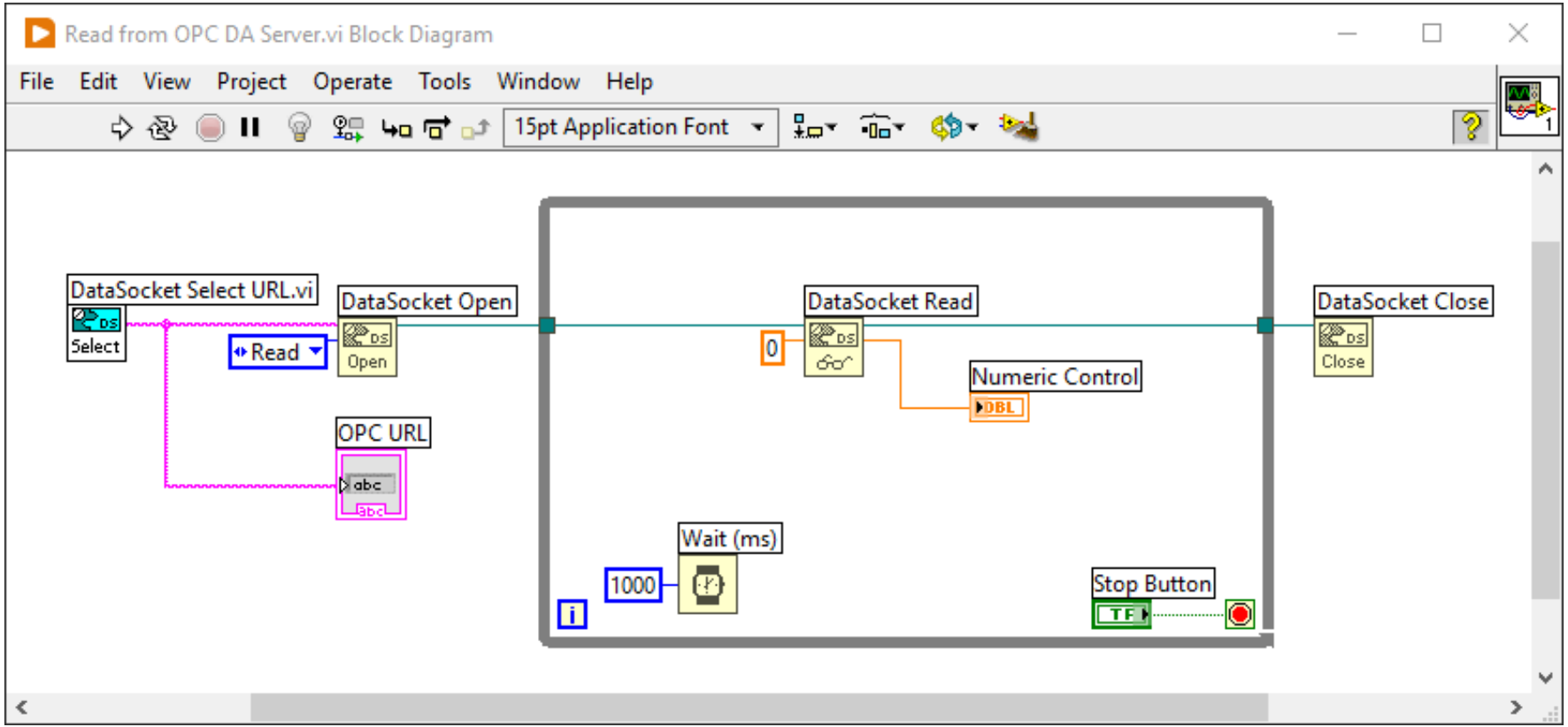
Write Data to OPC DA Server



Read Data from OPC DA Server



Read Data from OPC DA Server



MATLAB + Industrial Communication Toolbox

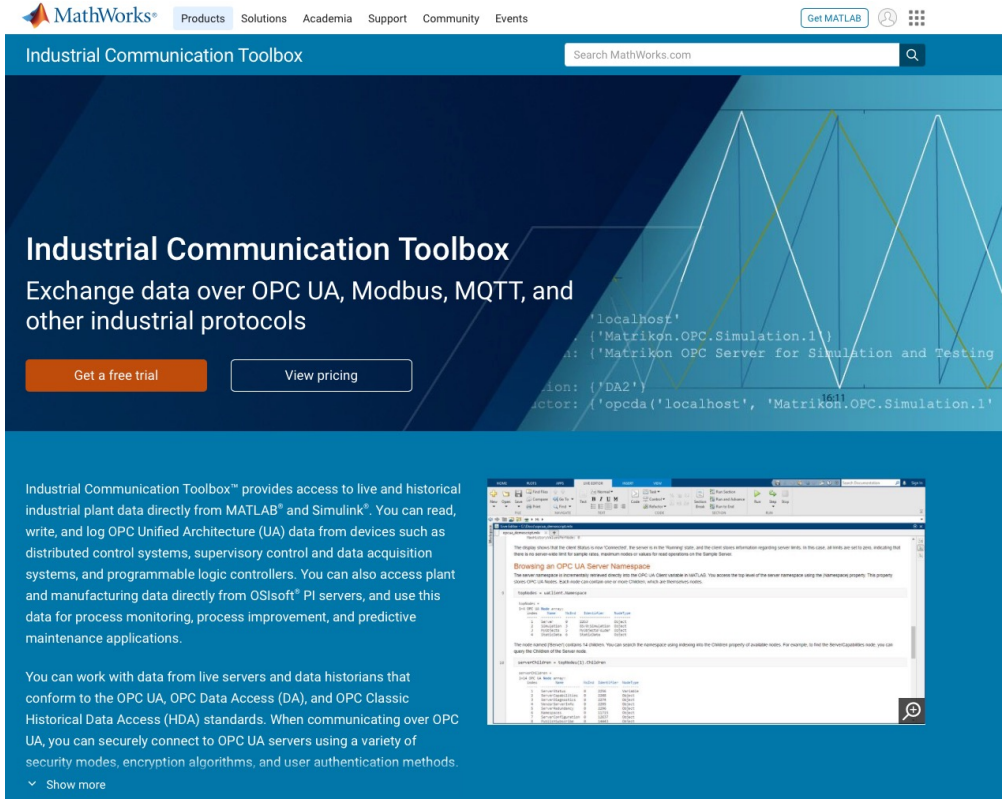
Hans-Petter Halvorsen

Table of Contents

OPC with MATLAB

- In order to use OPC with MATLAB you can use the "**Industrial Communication Toolbox**".
- The "Industrial Communication Toolbox" supports the following Protocols:
  - OPC, both OPC DA and OPC UA (previously "OPC Toolbox")
  - MQTT
  - Modbus
- **Note!** "Industrial Communication Toolbox" is a new Toolbox that is included in "MATLAB R2022a" and newer versions

Industrial Communication Toolbox



MATLAB and the Industrial Communication Toolbox are made by MathWorks

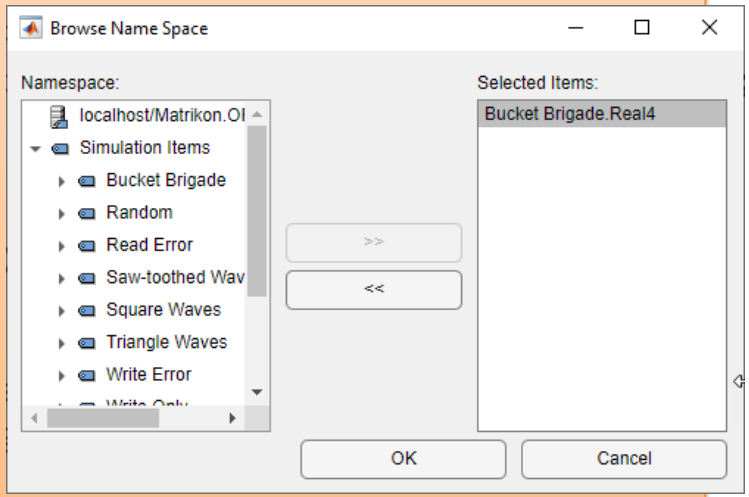
https://mathworks.com

https://mathworks.com/products/industrial-communication.html

MATLAB OPC DA

```
% Connect to OPC Server
da = opcda('localhost', 'Matrikon.OPC.Simulation.1');
connect(da);

% Create Group
grp = addgroup(da, 'DemoGroup');

%Add Tags
ItmList = browsenamespace(da);
itm = additem(grp, ItmList);

% Retrieve Data
data = read(grp);
opcdata = data.Value

%Clean Up
disconnect(da)
delete(da)
```

This Example makes it possible to select one or more OPC Tag from a pop-up window



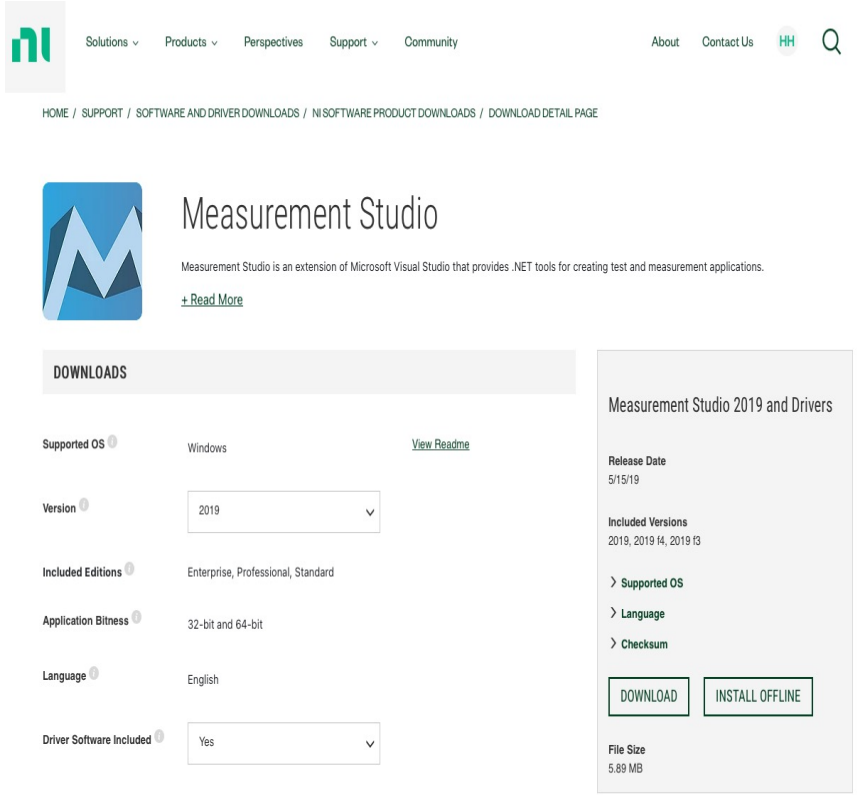
https://www.halvorsen.blog

Visual Studio/C# + Measurement Studio

Hans-Petter Halvorsen

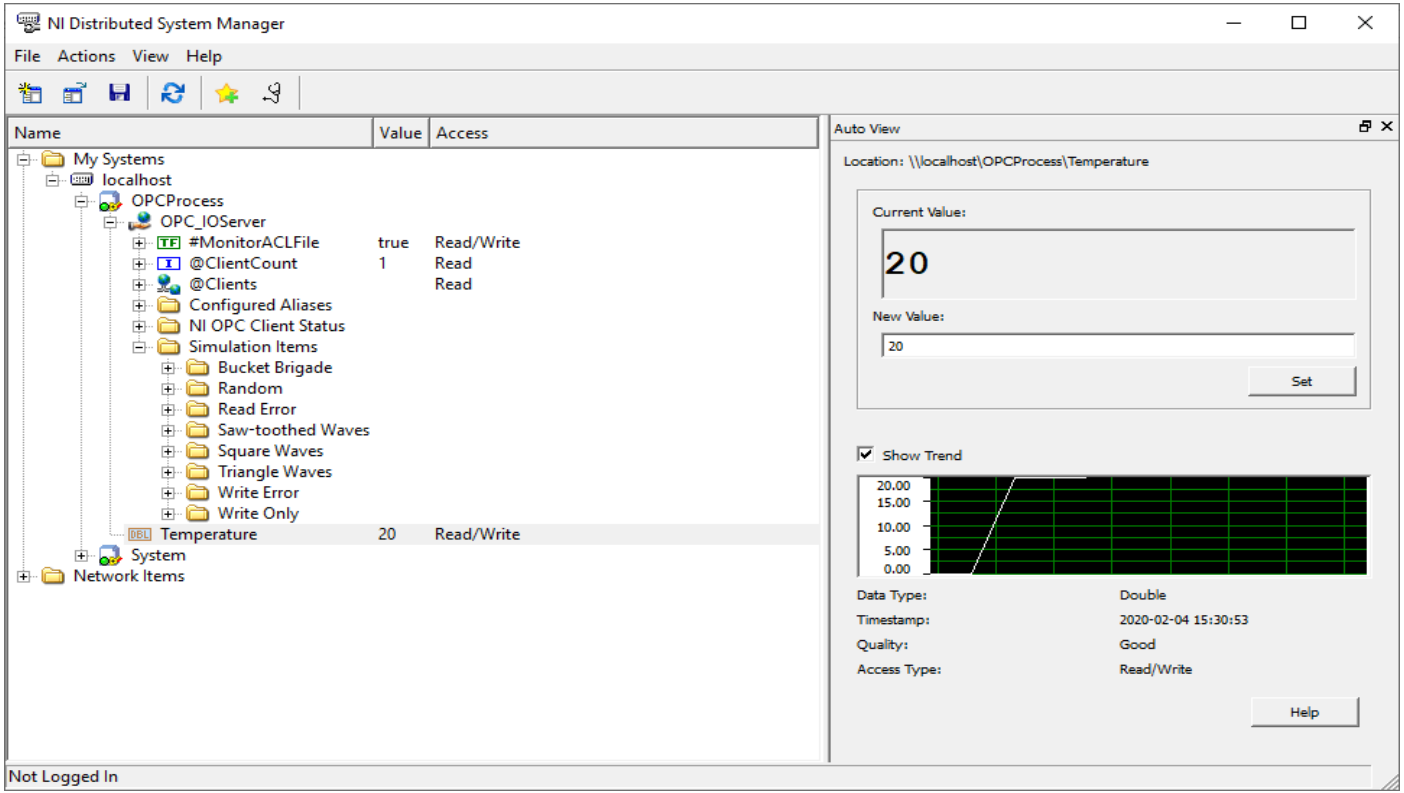
Table of Contents

Measurement Studio



- Measurement Studio (MS) is an Add-on package to Visual Studio created by NI (previously “National Instruments”)
- Same vendor as LabVIEW
- Makes it possible to communicate with an OPC DA Server from Visual Studio Code
- Uses the DataSocket Library (same as in LabVIEW)

NI Distributed System Manager



In order to configure the OPC Item to be used with Measurement Studio, we need to use NI Distributed System Manager

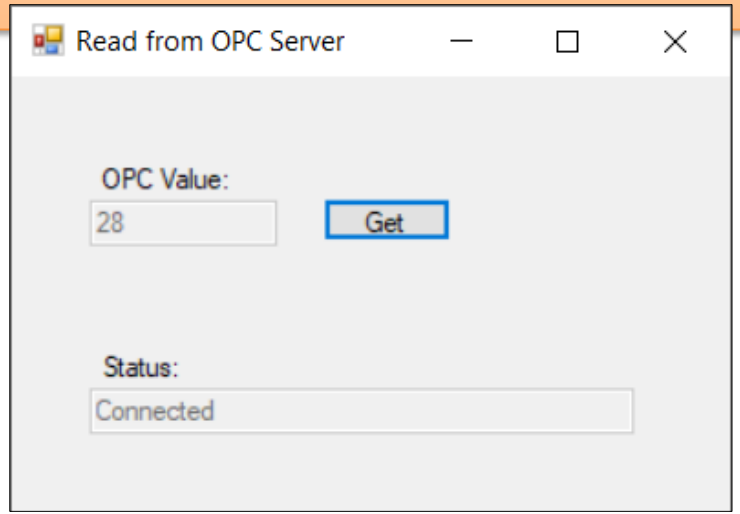
MS OPC Example

```
using NationalInstruments.NetworkVariable;
```

```
private NetworkVariableReader<double> _reader;
private const string NetworkVariableLocation = @"\\localhost\OPCProcess\Temperature";

public Form1()
{
    InitializeComponent();
    ConnectOPCServer();
}
```

MS OPC Read Example

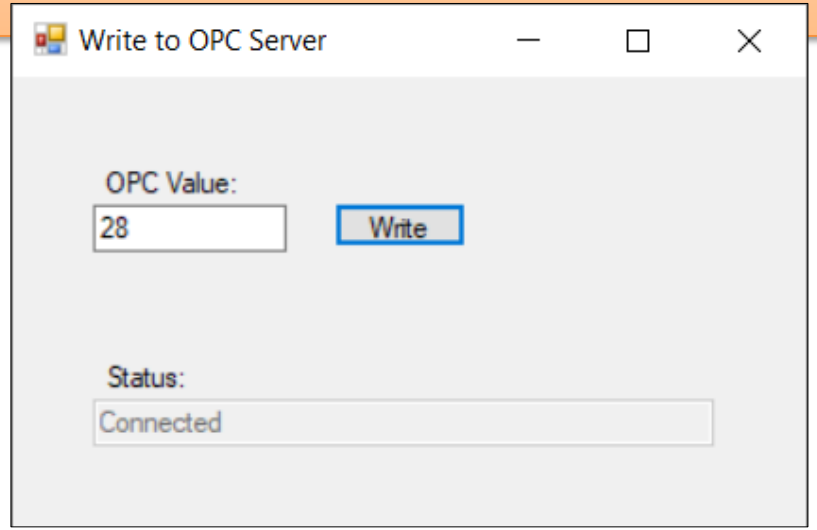
```
private void ConnectOPCServer()
{
    _reader = new NetworkVariableReader<double>(NetworkVariableLocation);

    _reader.Connect();

    txtStatus.Text = _reader.ConnectionStatus.ToString();
}
```




```
private void btnGetData_Click(object sender, EventArgs e)
 {
     NetworkVariableData<double> opcdata = null;
      try
       {
          opcdata = _reader.ReadData();
          txtOpcData.Text = opcdata.GetValue().ToString();
       }
       catch (TimeoutException)
       {
          MessageBox.Show("The read has timed out.", "Timeout");
          return;
       }
}
```

MS OPC Write Example

```
private void ConnectOPCServer()
{
    _writer = new NetworkVariableWriter<double>(NetworkVariableLocation);

    _writer.Connect();

     txtStatus.Text = _writer.ConnectionStatus.ToString();
}
```




```
private void btnWriteData_Click(object sender, EventArgs
{
    double temperature;
    try
    {
        temperature = Convert.ToDouble(txtOpcData.Text);

        _writer.WriteValue(temperature);
     }
     catch (TimeoutException)
     {
         MessageBox.Show("The read has timed out.", "Timeout");
         return;
     }
}
```

OPC UA

Hans-Petter Halvorsen

Table of Contents

OPC UA Servers

- "OPC UA Server Simulator"
  - "**OPC UA Server Simulator**" from "Integration Objects" is an OPC UA Demo/Test Server which you can download and use for free
- "LabVIEW OPC UA Server"
  - With "**LabVIEW OPC UA Toolkit**" you can create your own OPC UA Server
- Lots of other alternatives …

https://www.halvorsen.blog

OPC UA
Demo/Test Software

Hans-Petter Halvorsen

Table of Contents

OPC UA Demo/Test Software

- OPC UA Server
  - E.g., "**OPC UA Server Simulator**" from "Integration Objects", which is an OPC UA Demo/Test Server which you can download and use for free
- OPC UA Client
  - E.g., "**OPC UA Client**" from "Integration Objects", which is a free client tool that supports the main OPC Unified Architecture information models.

OPC UA Server Simulator

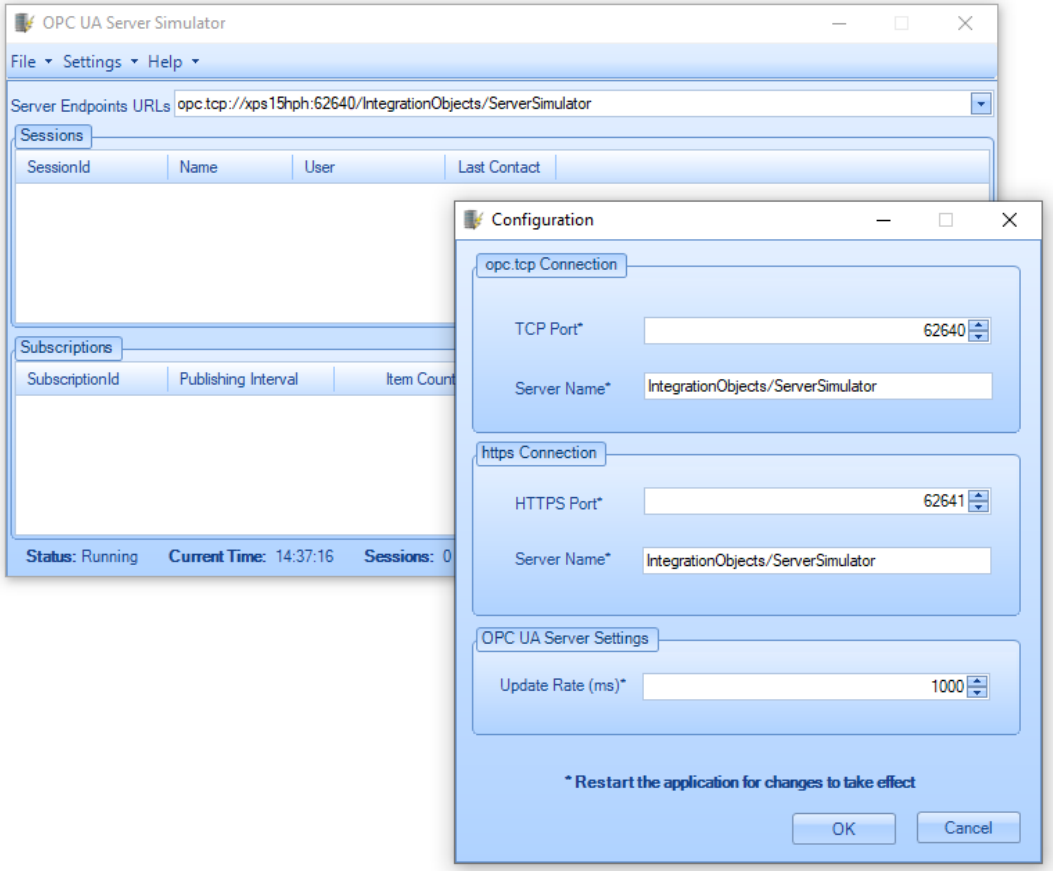
Hans-Petter Halvorsen

Table of Contents

OPC UA Server Simulator

https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-server-simulator/

OPC UA Server Simulator

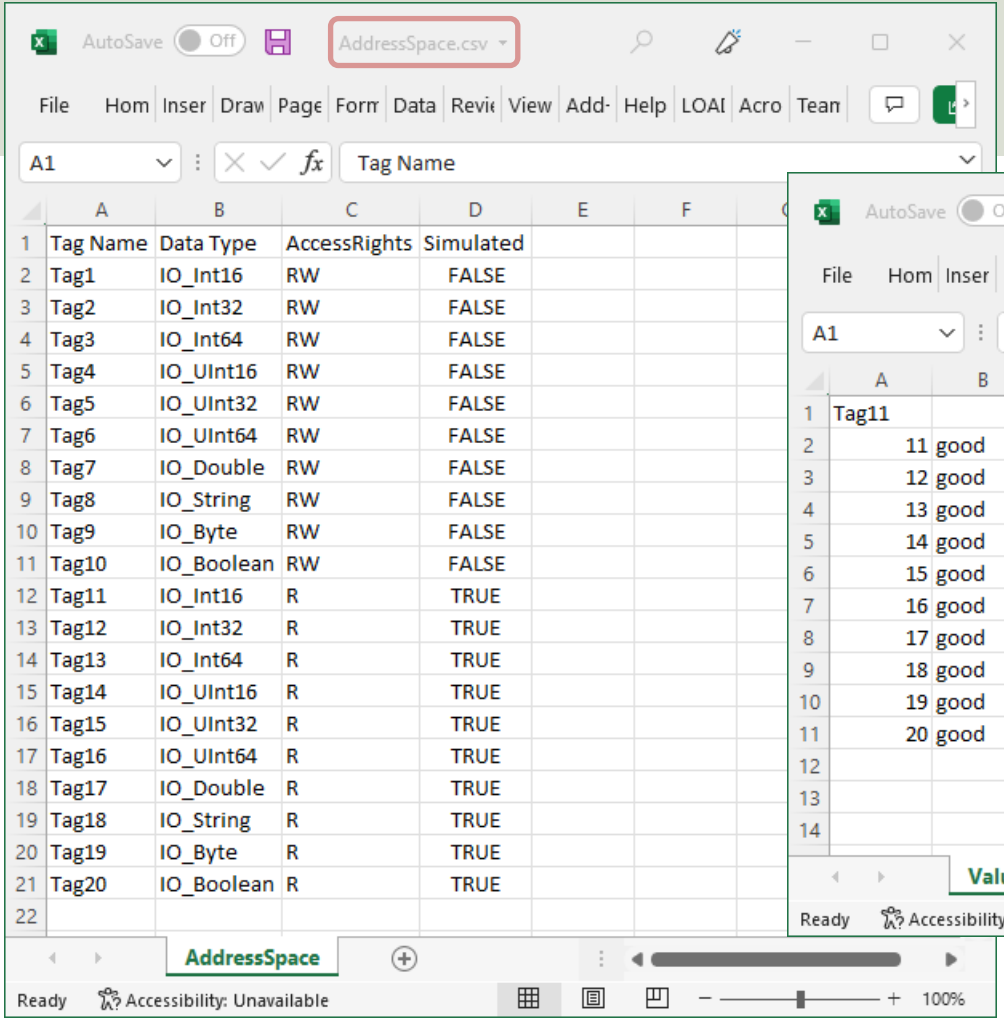


For the OPC UA Part we will use the “OPC UA Server Simulator”, which is an OPC UA Demo/Test Server which you can download and use for free

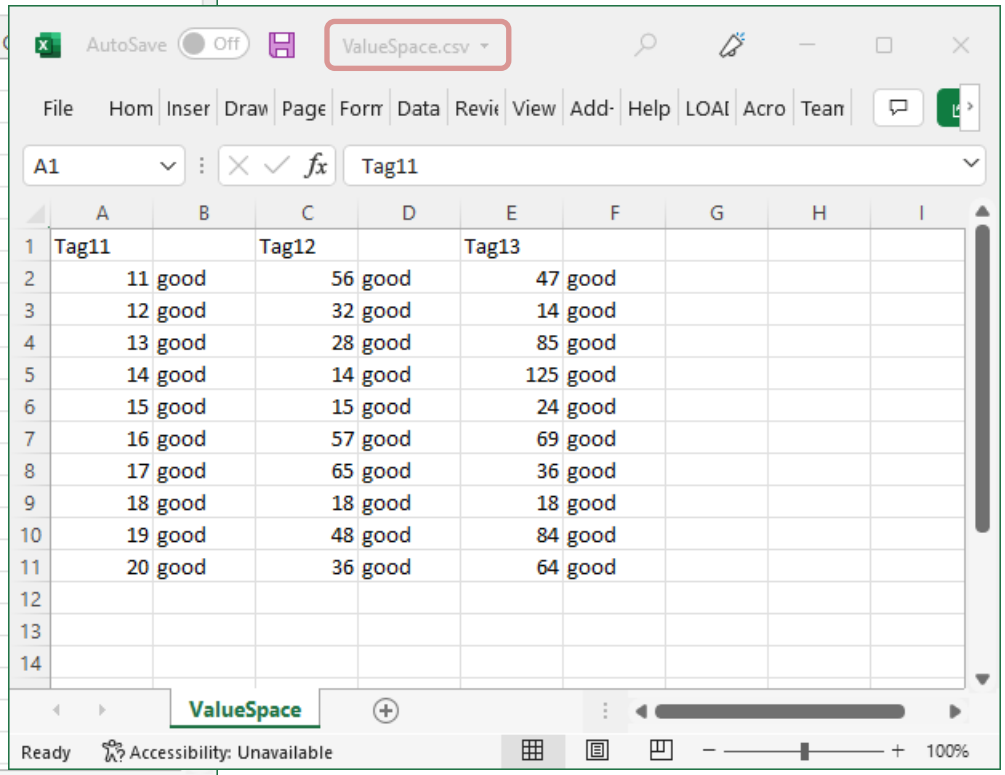
OPC UA Server Simulator

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



	A	B	C	D
1	Tag Name	Data Type	AccessRights	Simulated
2	Tag1	IO_Int16	RW	FALSE
3	Tag2	IO_Int32	RW	FALSE
4	Tag3	IO_Int64	RW	FALSE
5	Tag4	IO_UInt16	RW	FALSE
6	Tag5	IO_UInt32	RW	FALSE
7	Tag6	IO_UInt64	RW	FALSE
8	Tag7	IO_Double	RW	FALSE
9	Tag8	IO_String	RW	FALSE
10	Tag9	IO_Byte	RW	FALSE
11	Tag10	IO_Boolean	RW	FALSE
12	Tag11	IO_Int16	R	TRUE
13	Tag12	IO_Int32	R	TRUE
14	Tag13	IO_Int64	R	TRUE
15	Tag14	IO_UInt16	R	TRUE
16	Tag15	IO_UInt32	R	TRUE
17	Tag16	IO_UInt64	R	TRUE
18	Tag17	IO_Double	R	TRUE
19	Tag18	IO_String	R	TRUE
20	Tag19	IO_Byte	R	TRUE
21	Tag20	IO_Boolean	R	TRUE
22				



	A	B	C	D	E	F	G	H	I
1	Tag11		Tag12		Tag13				
2	11	good	56	good	47	good			
3	12	good	32	good	14	good			
4	13	good	28	good	85	good			
5	14	good	14	good	125	good			
6	15	good	15	good	24	good			
7	16	good	57	good	69	good			
8	17	good	65	good	36	good			
9	18	good	18	good	18	good			
10	19	good	48	good	84	good			
11	20	good	36	good	64	good			
12									
13									
14									

“OPC UA Client”

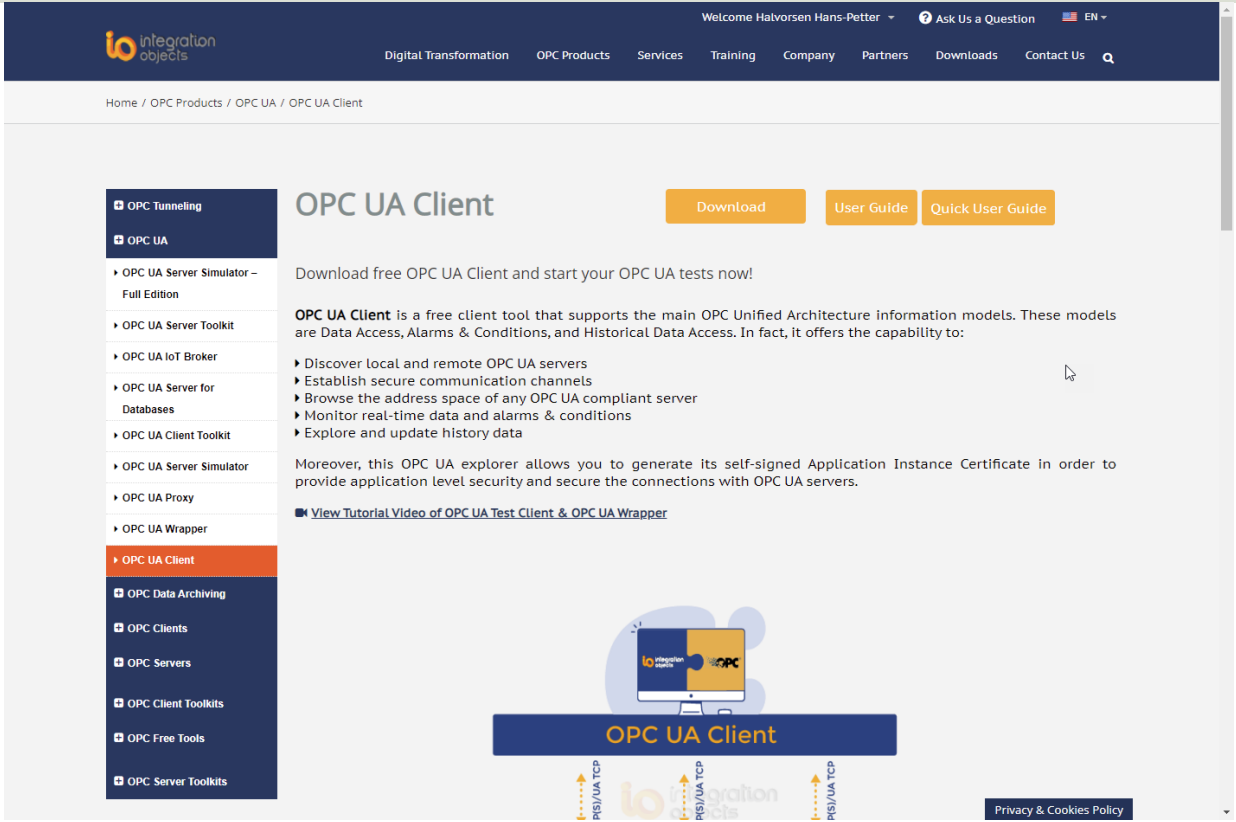
Hans-Petter Halvorsen

Table of Contents

“OPC UA Client”

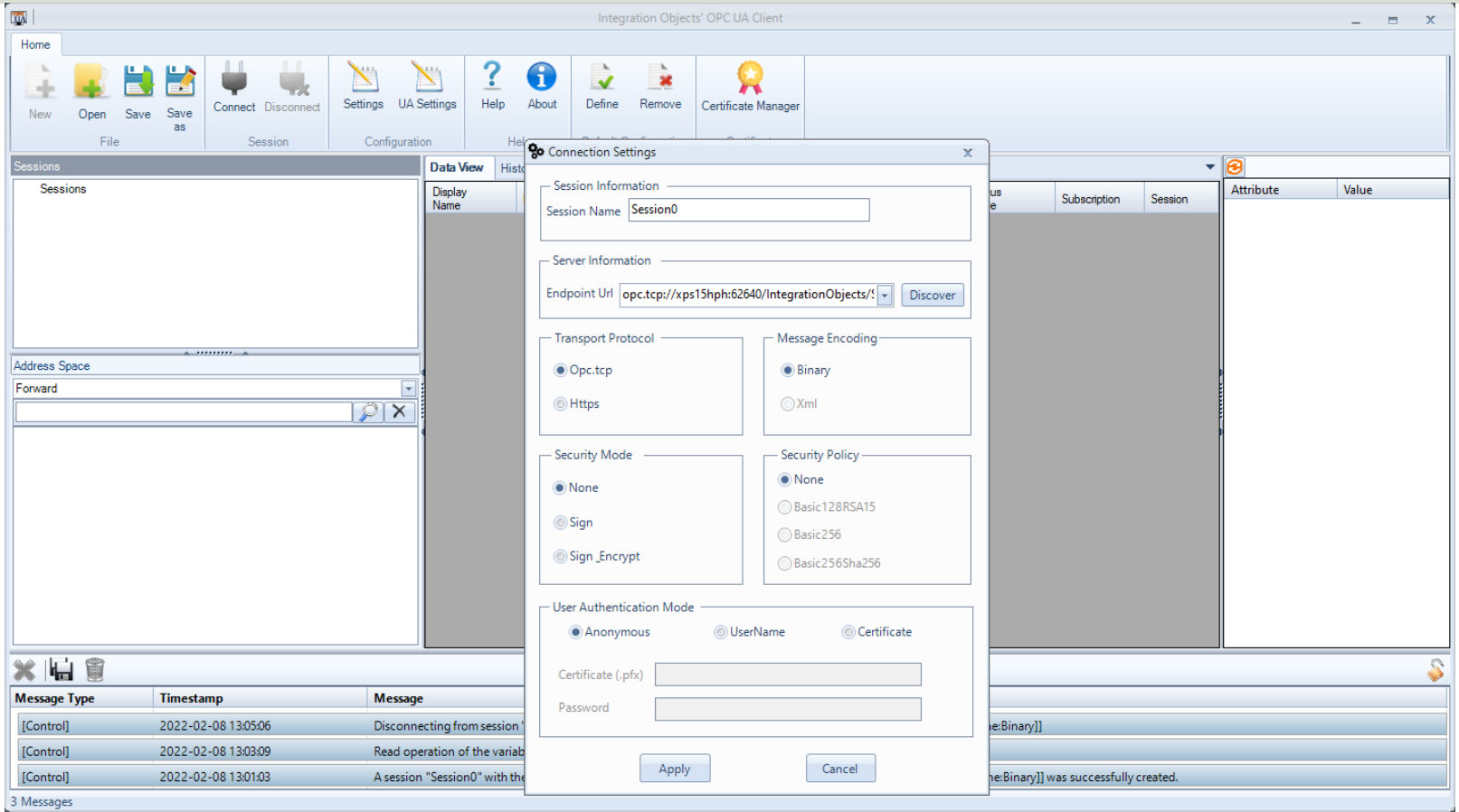
- “OPC UA Client” is a free OPC client tool that supports the main OPC Unified Architecture information models.
- These models are Data Access, Alarms & Conditions, and Historical Data Access
- Handy to use to test OPC communication
- https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-client/

“OPC UA Client”

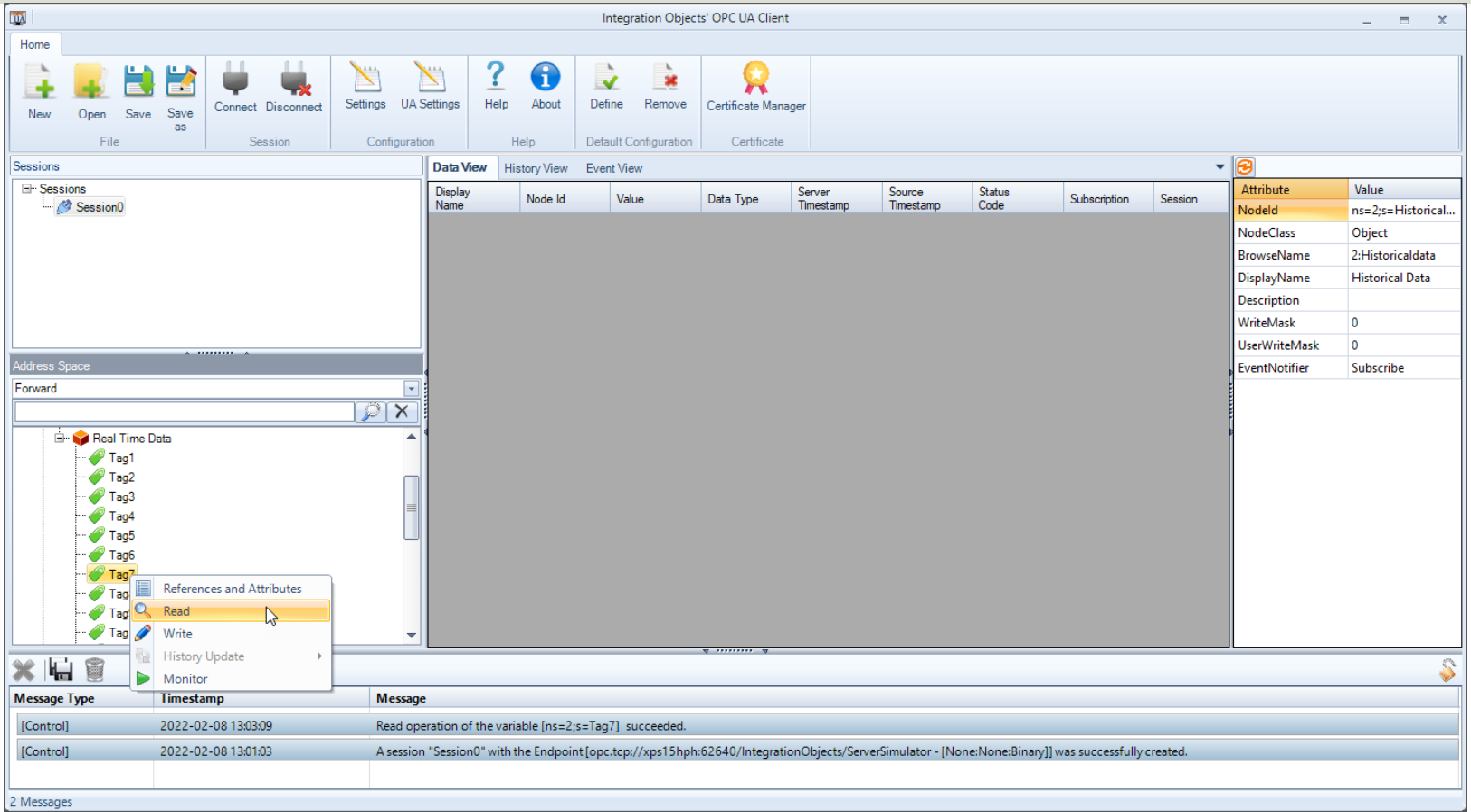


https://integrationobjects.com/sioth-opc/sioth-opc-unified-architecture/opc-ua-client/

“OPC UA Client”



“OPC UA Client”



https://www.halvorsen.blog

OPC UA Programming Tools

Hans-Petter Halvorsen

Table of Contents

OPC UA Programming Tools

Software and Programming Tools that can be used for communicating with OPC UA Servers:

- LabVIEW + LabVIEW OPC UA Toolkit
- MATLAB + Industrial Communication Toolkit (supports both OPC DA and UA)
- Visual Studio/C# + “OPC UA .NET SDK”

Many other alternatives exists

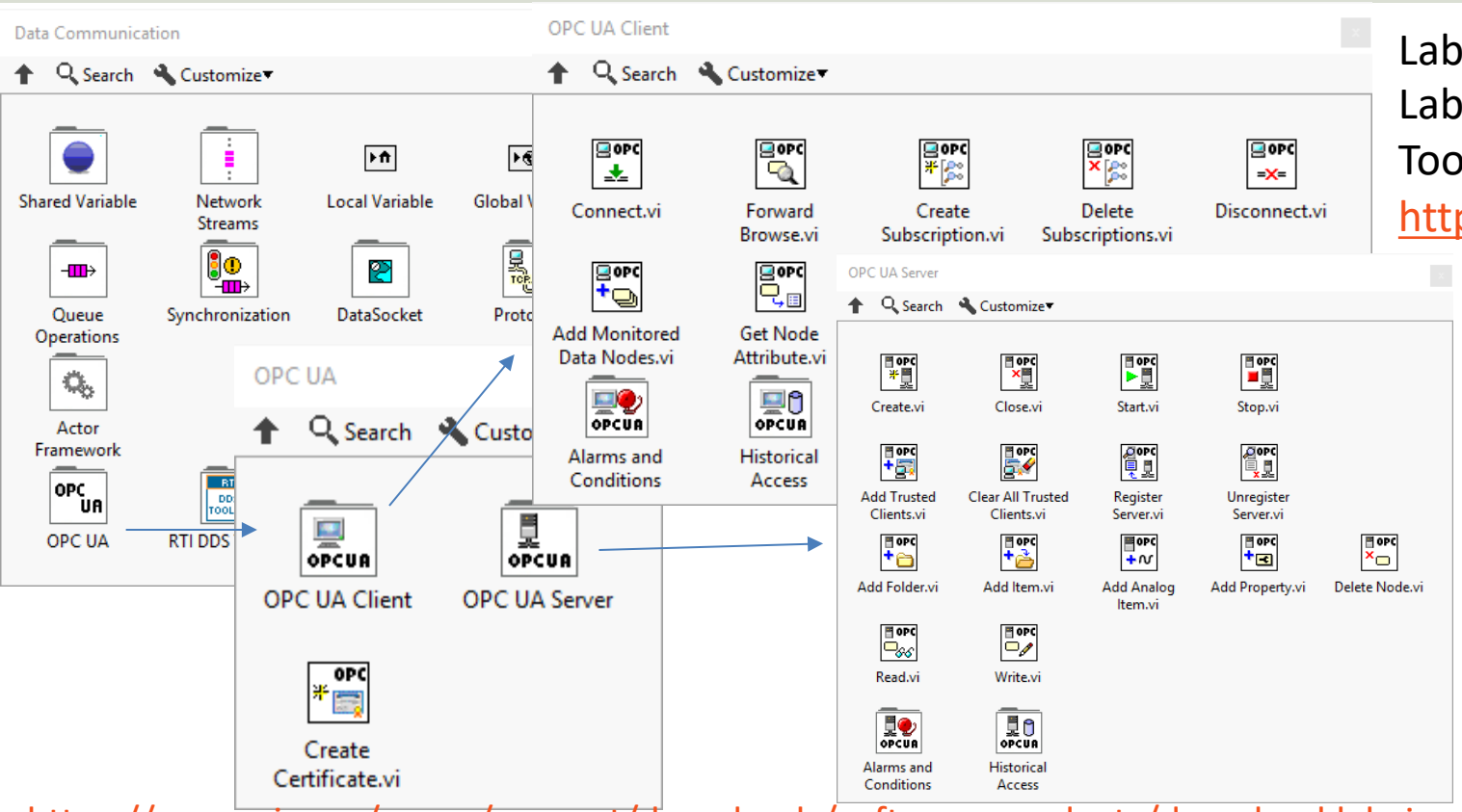
https://www.halvorsen.blog

LabVIEW OPC UA Toolkit

Hans-Petter Halvorsen

Table of Contents

LabVIEW OPC UA Toolkit

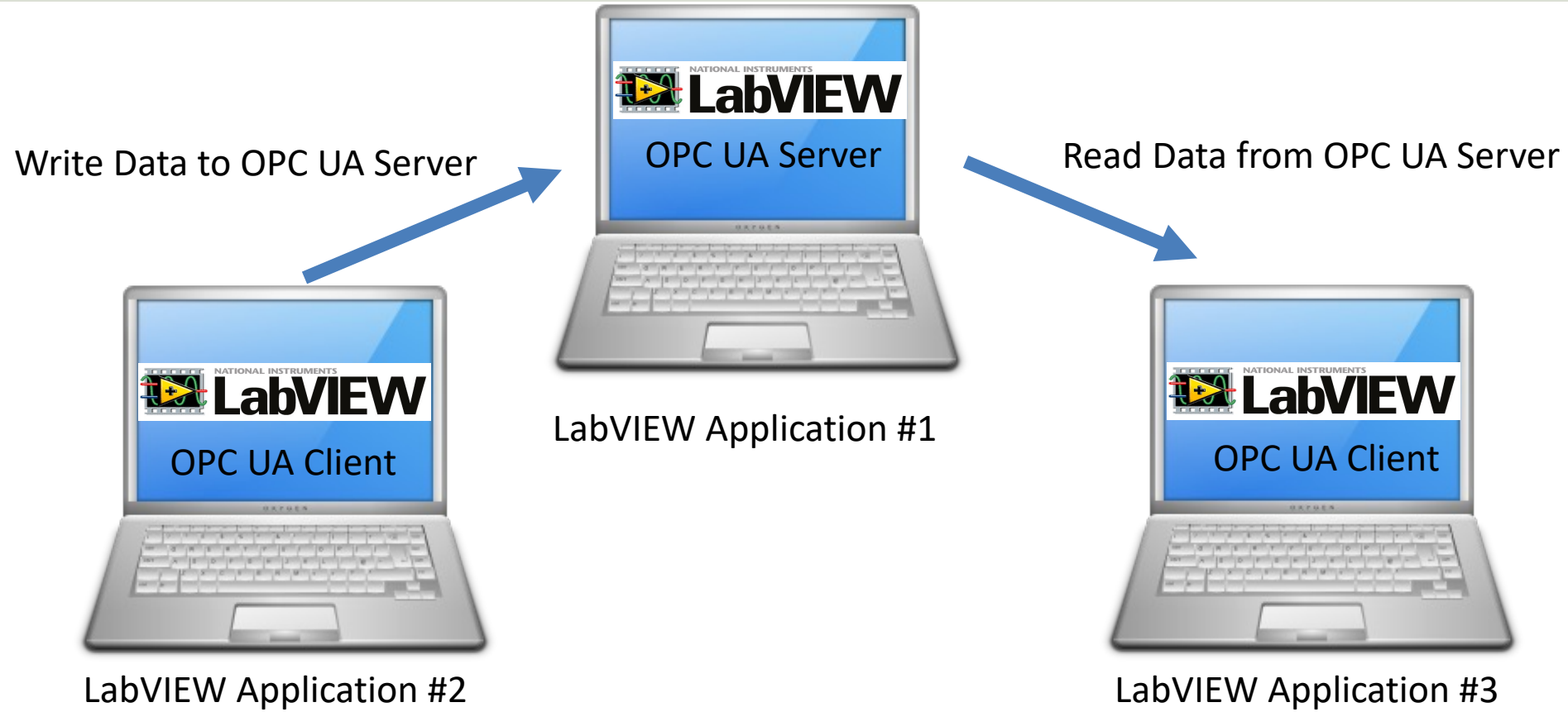


LabVIEW and the LabVIEW OPC UA Toolkit are made by NI

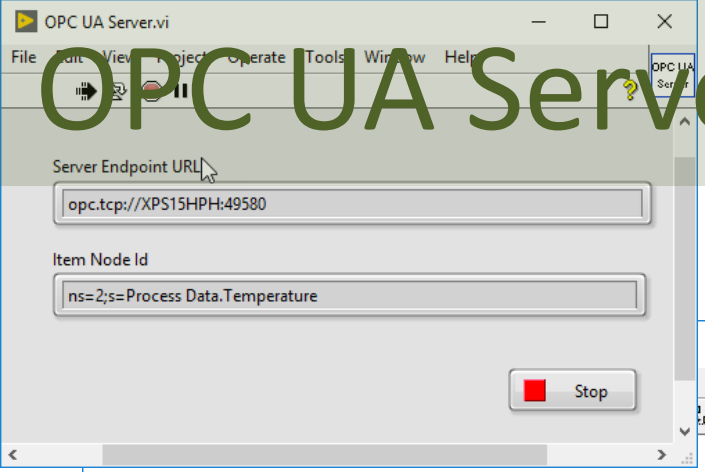
https://www.ni.com

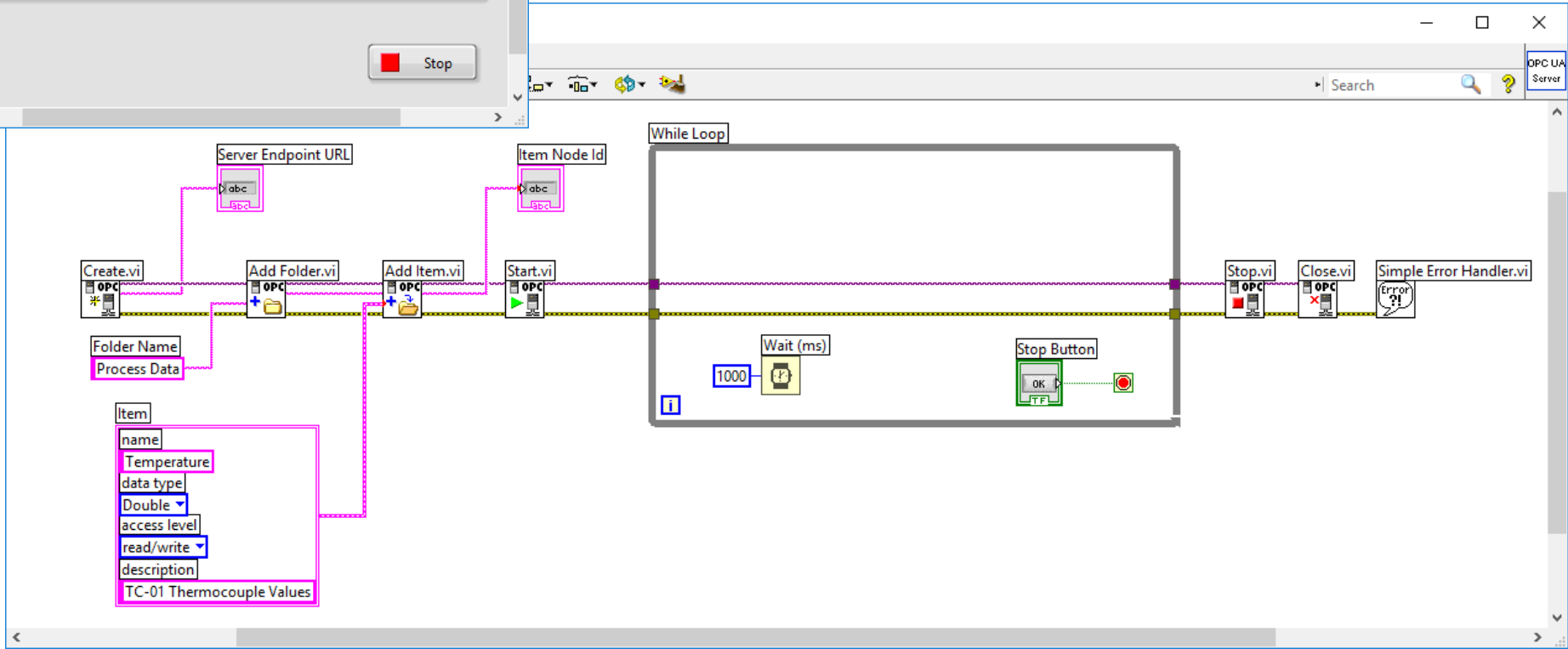
https://www.ni.com/en-no/support/downloads/software-products/download.labview-opc-ua-toolkit.html

OPC UA in LabVIEW

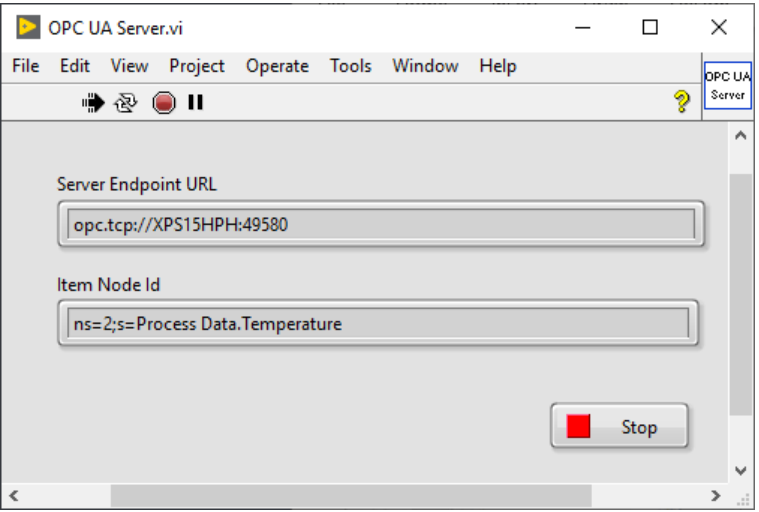


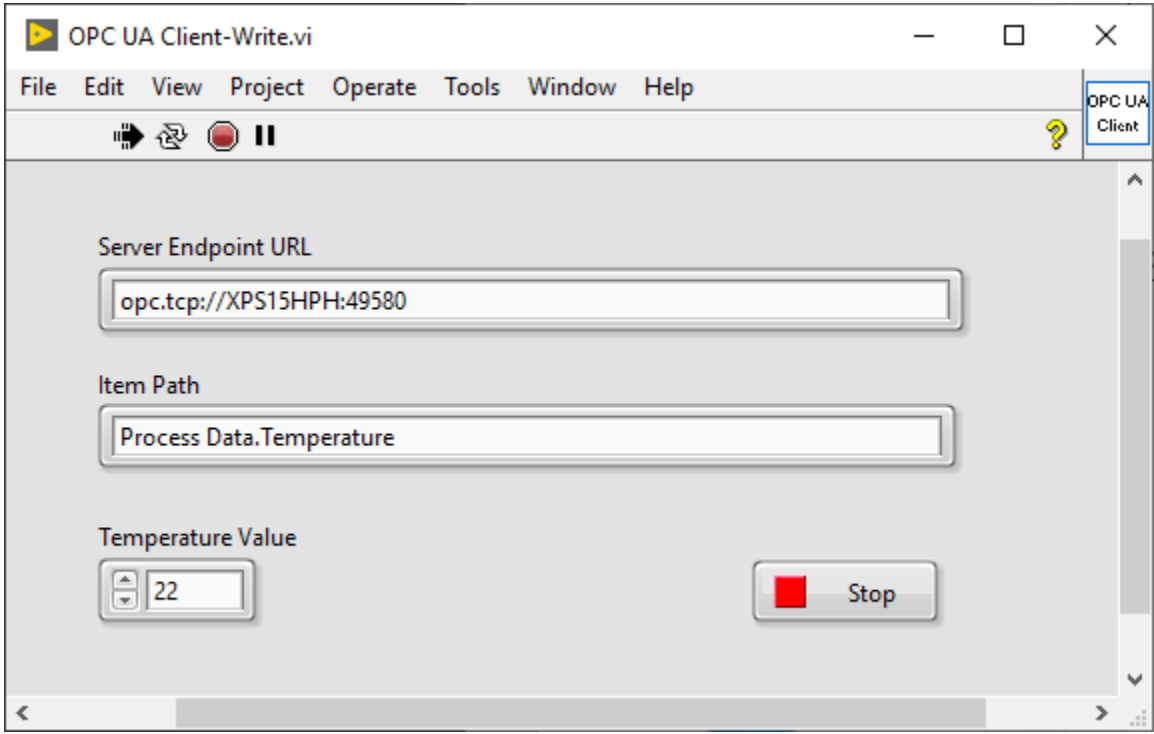
OPC UA Server Example in LabVIEW



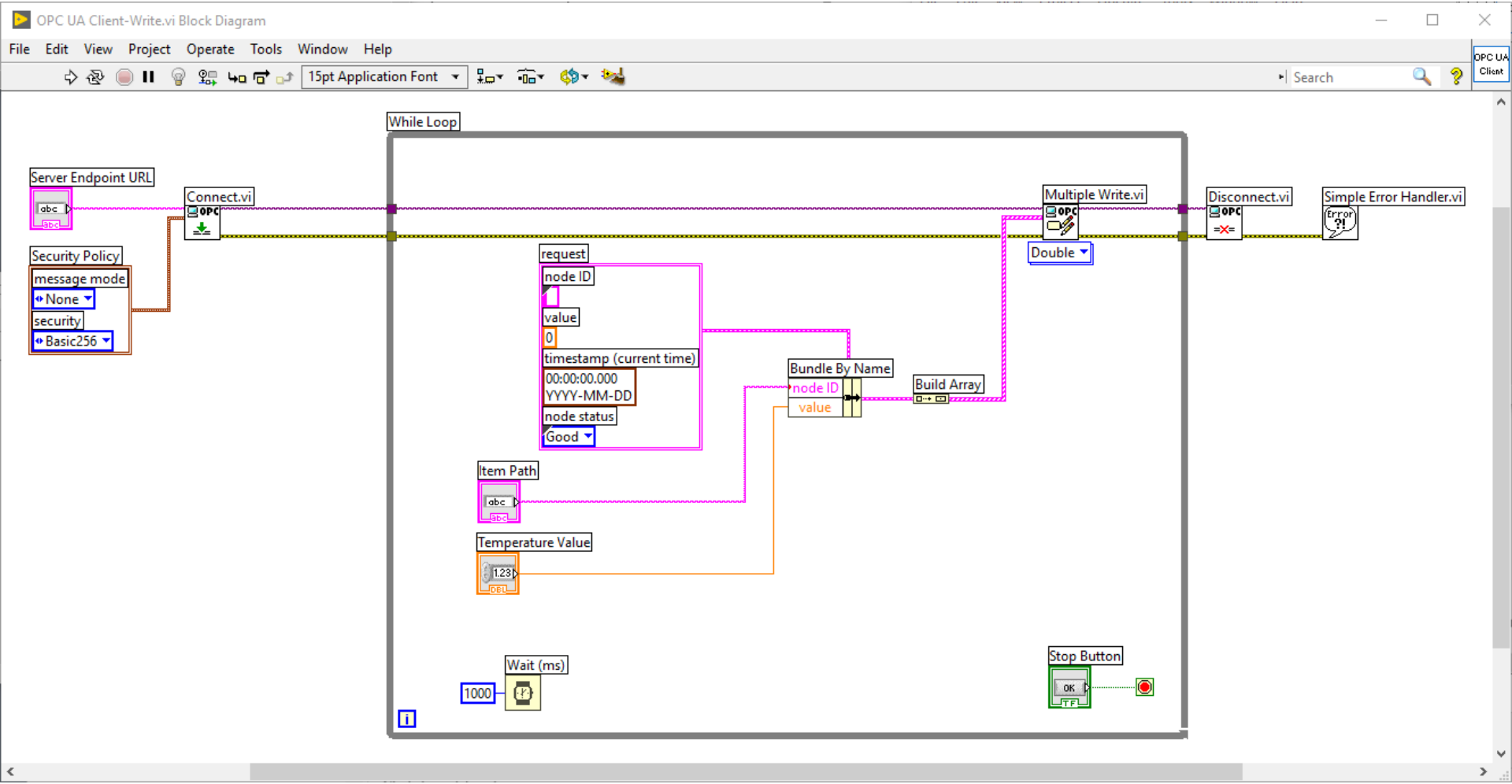


OPC UA Client Write Data

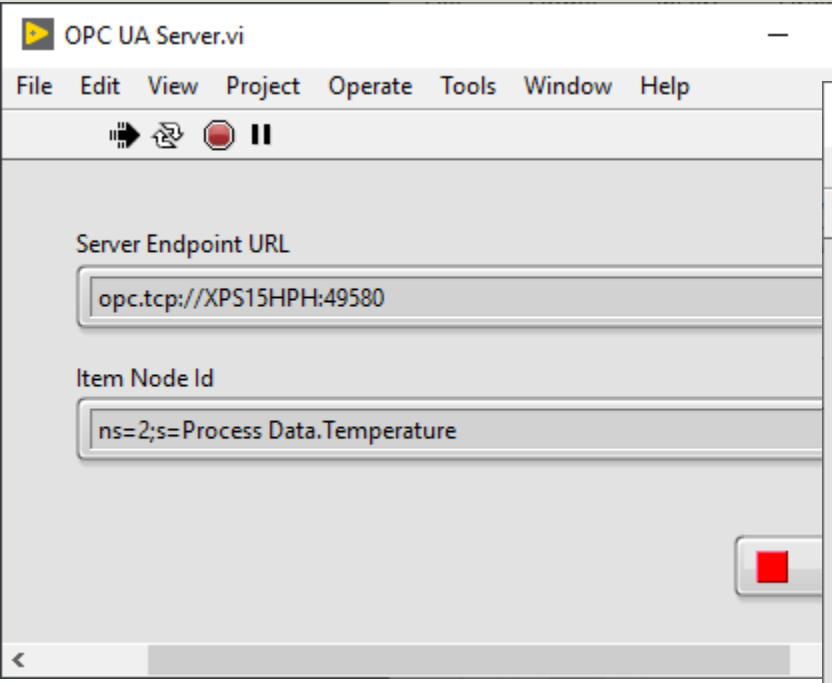


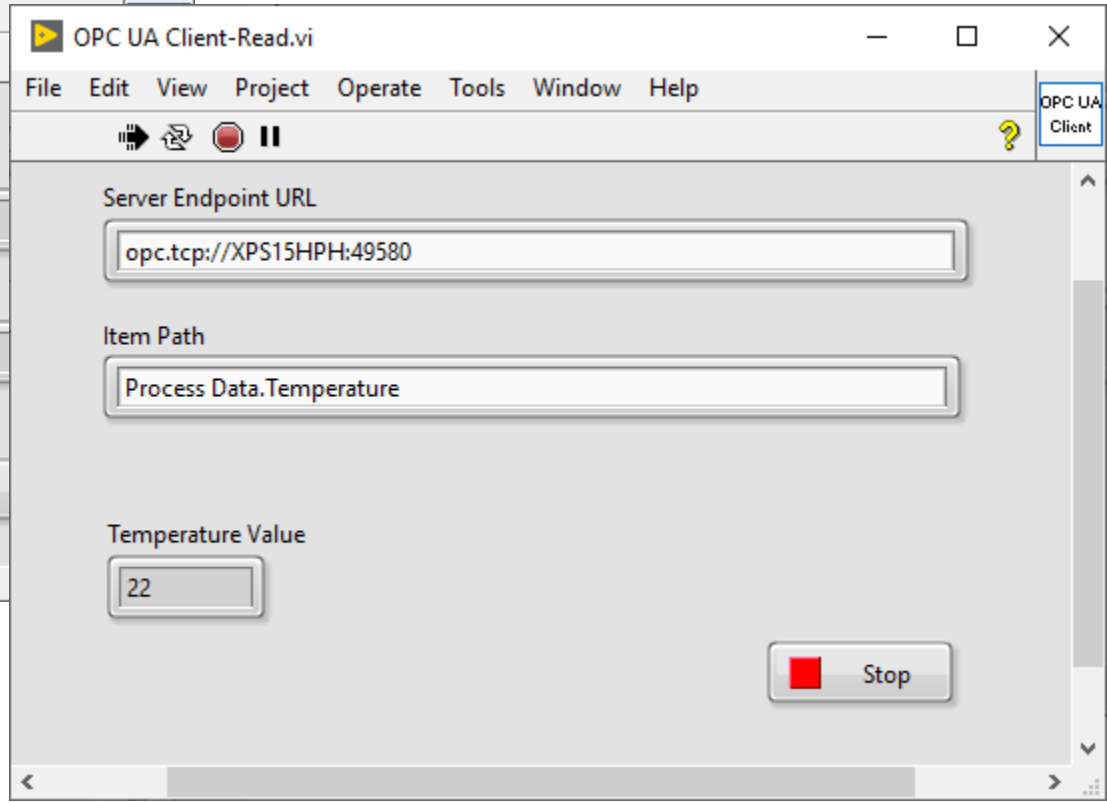


Using **OPC UA Toolkit** with LabVIEW 2017 or newer

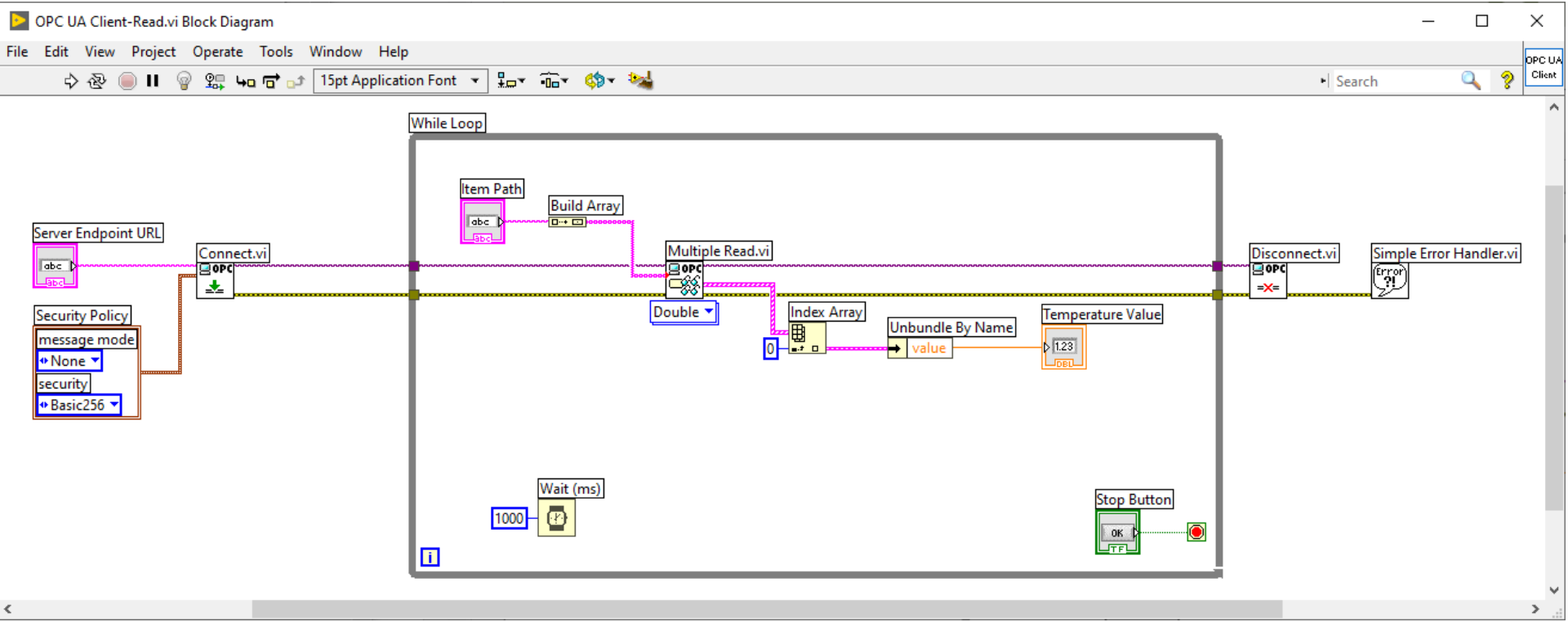

OPC UA Client-Write.vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
While Loop
Server Endpoint URL
Connect.vi
Security Policy
message mode
None
security
Basic256
request
node ID
value
0
timestamp (current time)
00:00:00.000
YYYY-MM-DD
node status
Good
Item Path
Temperature Value
Bundle By Name
node ID
value
Build Array
Multiple Write.vi
Double
Disconnect.vi
Simple Error Handler.vi
Wait (ms)
1000
Stop Button

OPC UA Client Read Data





Block Diagram



MATLAB Industrial Communication Toolbox

Hans-Petter Halvorsen

Table of Contents

OPC with MATLAB

- In order to use OPC with MATLAB you can use the "**Industrial Communication Toolbox**".
- The "Industrial Communication Toolbox" supports the following Protocols:
  - OPC, both OPC DA and OPC UA (previously "OPC Toolbox")
  - MQTT
  - Modbus
- **Note!** "Industrial Communication Toolbox" is a new Toolbox that is included in "MATLAB R2022a" and newer versions

Industrial Communication Toolbox

- The Industrial Communication Toolbox supports:
  - OPC DA
  - OPC HDA
  - OPC UA
- Resources:
  - https://mathworks.com/products/industrial-communication.html
  - https://mathworks.com/help/icomm/data-reading-and-writing.html

OPC UA – Read Data

1. Locate Your OPC UA Server
```
serverList = opcuaserverinfo('localhost')
```
2. Create an OPC UA Client
```
uaClient = opcua('localhost', port)
```
3. Connect to the Server
```
connect(uaClient)
```
4. Browse OPC UA Server Namespace
```
serverNodes = browseNamespace(uaClient)
```
5. Read Current Values from the OPC UA Server
```
[val,ts,qual] = readValue(uaClient,serverNodes)
```
6. Disconnect
```
disconnect(uaClient)
```

OPC UA – Read Data

```
clear, clc

uaClient = opcua('localhost', 62640)

connect(uaClient)

serverNodes = browseNamespace(uaClient)

[val,ts,qual] = readValue(uaClient,serverNodes)

disconnect(uaClient);
```

OPC UA – Read Data 2

```
clear, clc

uaClient = opcua('localhost', 62640)

connect(uaClient)
topNodes = uaClient.Namespace
serverChildren = topNodes(4).Children
findNode = findNodeByName(topNodes,'Tag7', '-once')

opcNode = opcuanode(2,'Tag7',uaClient)
[value,timestamp,quality] = readValue(uaClient,opcNode)

disconnect(uaClient);
```

Read Data

Here you don't need to select the Tag from the "Browse Name Space" window every time

OPC UA – Write Data

1. Locate Your OPC UA Server
```
serverList = opcuaserverinfo('localhost')
```
2. Create an OPC UA Client
```
uaClient = opcua('localhost', port)
```
3. Connect to the Server
```
connect(uaClient)
```
4. Browse OPC UA Server Namespace
```
serverNodes = browseNamespace(uaClient)
```
5. Write Current Values to the OPC UA Server
```
newValue = 22.5
writeValue(uaClient, serverNodes, newValue);
```
6. Disconnect
```
disconnect(uaClient)
```

OPC UA – Write Data

```
clear, clc

uaClient = opcua('localhost', 62640)

connect(uaClient)

serverNodes = browseNamespace(uaClient)

newValue = 21.7;
writeValue(uaClient, serverNodes, newValue);

[value,timestamp,quality] = readValue(uaClient,serverNodes)

disconnect(uaClient);
```

Visual Studio/C# + “OPC UA .NET SDK”

Hans-Petter Halvorsen

Table of Contents

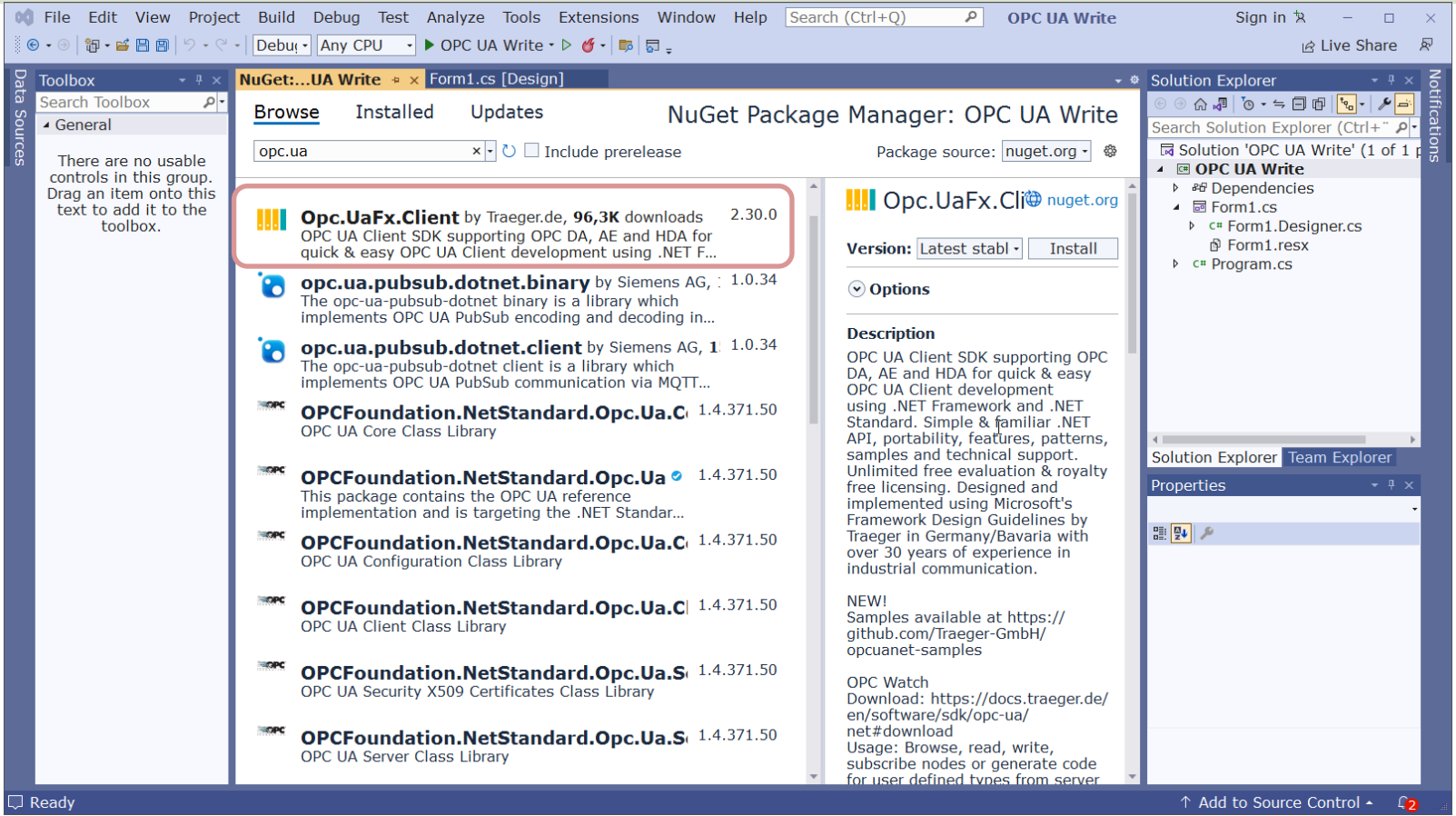
OPC UA with Visual Studio/C#

- 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
  - It can be used in “Evaluation Mode” for Test and Demo purposes

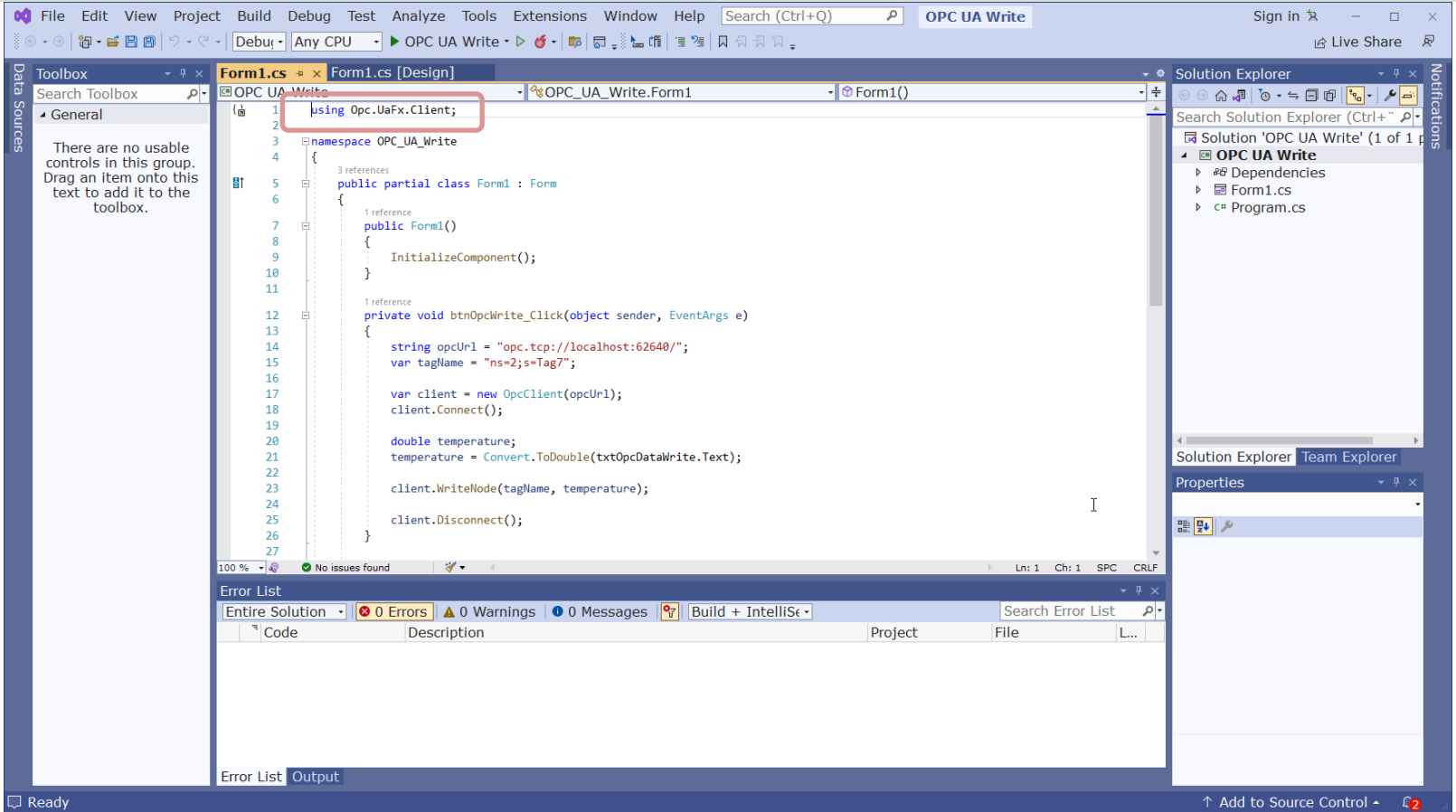
“OPC UA .NET SDK”

- The “OPC UA .NET SDK” comes with an evaluation license which can be used unlimited where each application runs 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

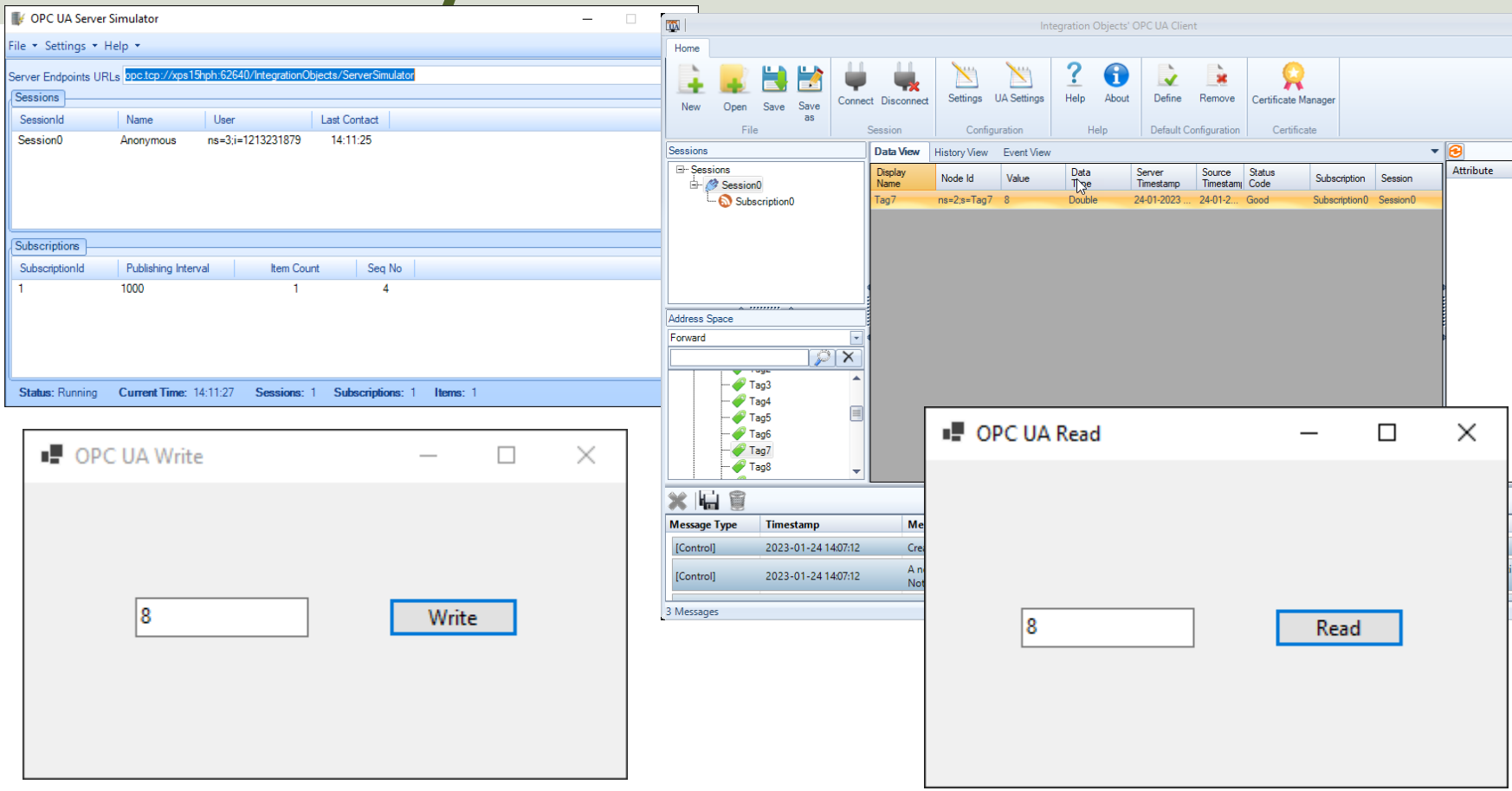
NuGet Package



Visual Studio Project



System Overview



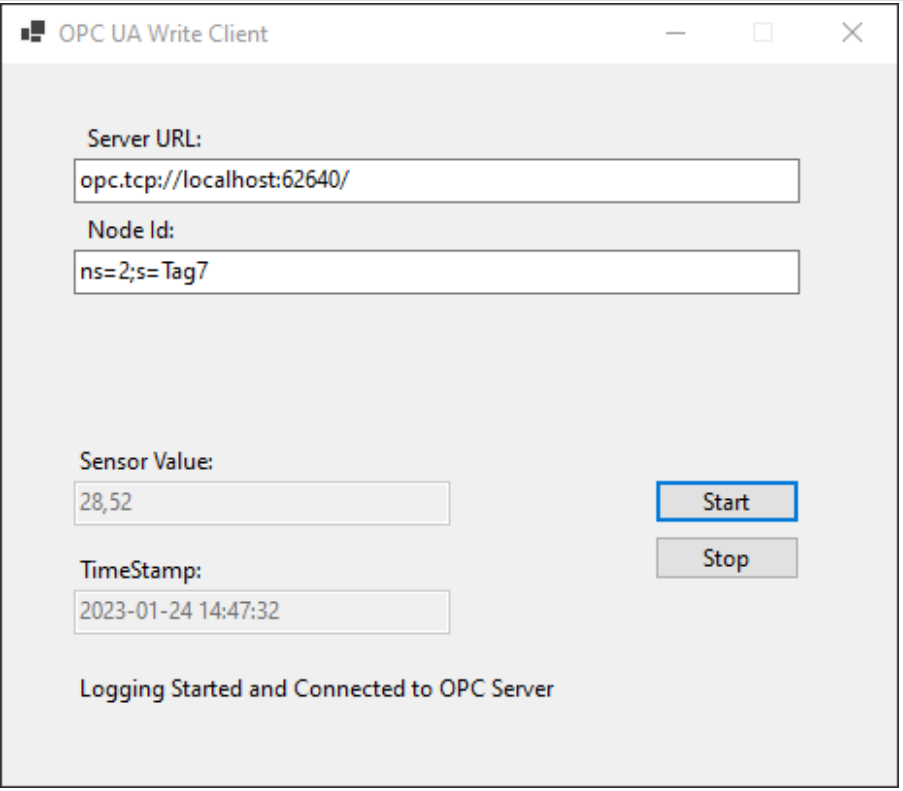
OPC UA Write

```
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();
}
```

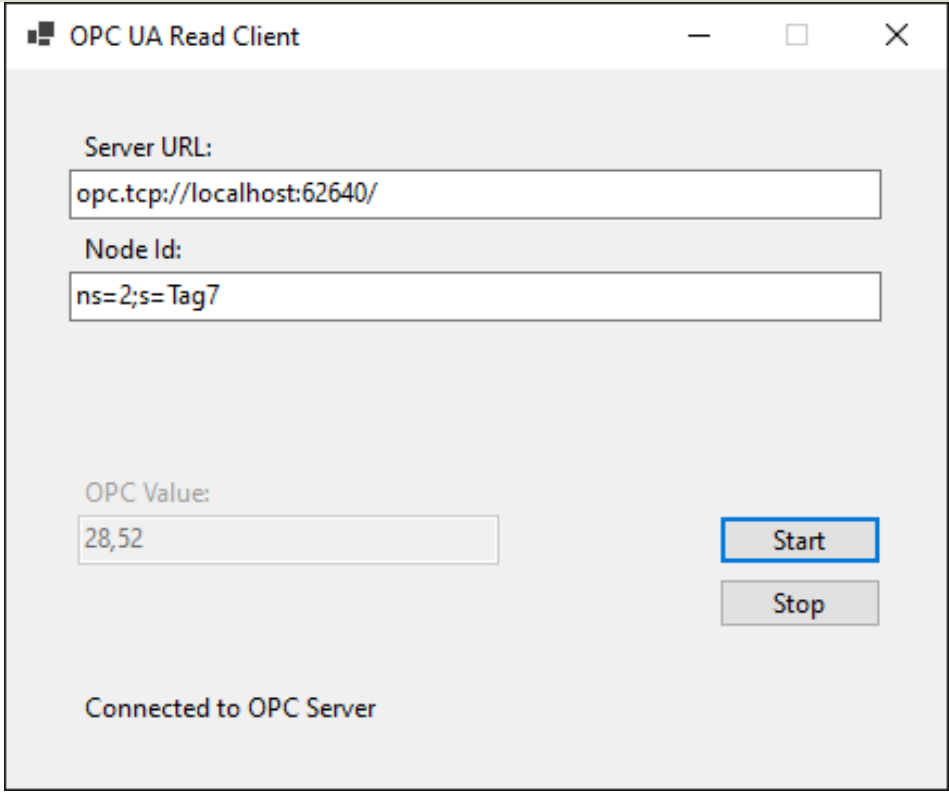
OPC UA Read

```
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();
}
```

Improved Example



OPC UA Write C# App



OPC UA Read C# App

Summary

- What is OPC?
- OPC DA
  - OPC DA Servers
    - MatrikonOPC Simulation Server
    - “OPC Server Simulators” from Integration Objects
    - NI OPC Servers
  - OPC DA Programming Tools
    - LabVIEW + DataSocket
    - MATLAB + Industrial Communication Toolbox
    - Visual Studio/C# + Measurement Studio
- OPC UA
  - OPC UA Demo/Test Software
    - “OPC UA Server Simulator” from Integration Objects
    - “OPC UA Client” from Integration Objects
  - OPC UA Programming Tools
    - LabVIEW + LabVIEW OPC UA Toolkit
    - MATLAB + Industrial Communication Toolbox
    - Visual Studio/C# + “OPC UA .NET SDK“ from Traeger

For all these Programming Languages and Packages, I have made separate Tutorials where I go through the development of the Applications and the Code in more details

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog