

<https://www.halvorsen.blog>



OPC with MATLAB

Industrial Communication Toolbox

Hans-Petter Halvorsen

Contents

- Introduction
- What is OPC?
- OPC with MATLAB
 - MATLAB Industrial Communication Toolbox
- OPC DA Examples
 - OPC DA – Example 1
 - OPC DA – Example 2
 - OPC DA – Example 3
- OPC UA Examples
 - OPC UA – Example 4
 - OPC UA – Example 5





Introduction

Hans-Petter Halvorsen

[Table of Contents](#)

Introduction

- In this Tutorial we will use **MATLAB** and the **Industrial Communication Toolbox** to communicate with different OPC Servers
- We will go through some examples where we use **OPC DA** and **OPC UA**

What do you need?

- OPC DA Server
 - E.g., “MatrikonOPC Simulation Server”, which is an OPC DA Demo/Test Server which you can download and use for free
- OPC UA Server
 - E.g., “OPC UA Server Simulator”, which is an OPC UA Demo/Test Server which you can download and use for free
- MATLAB
- Industrial Communication Toolkit

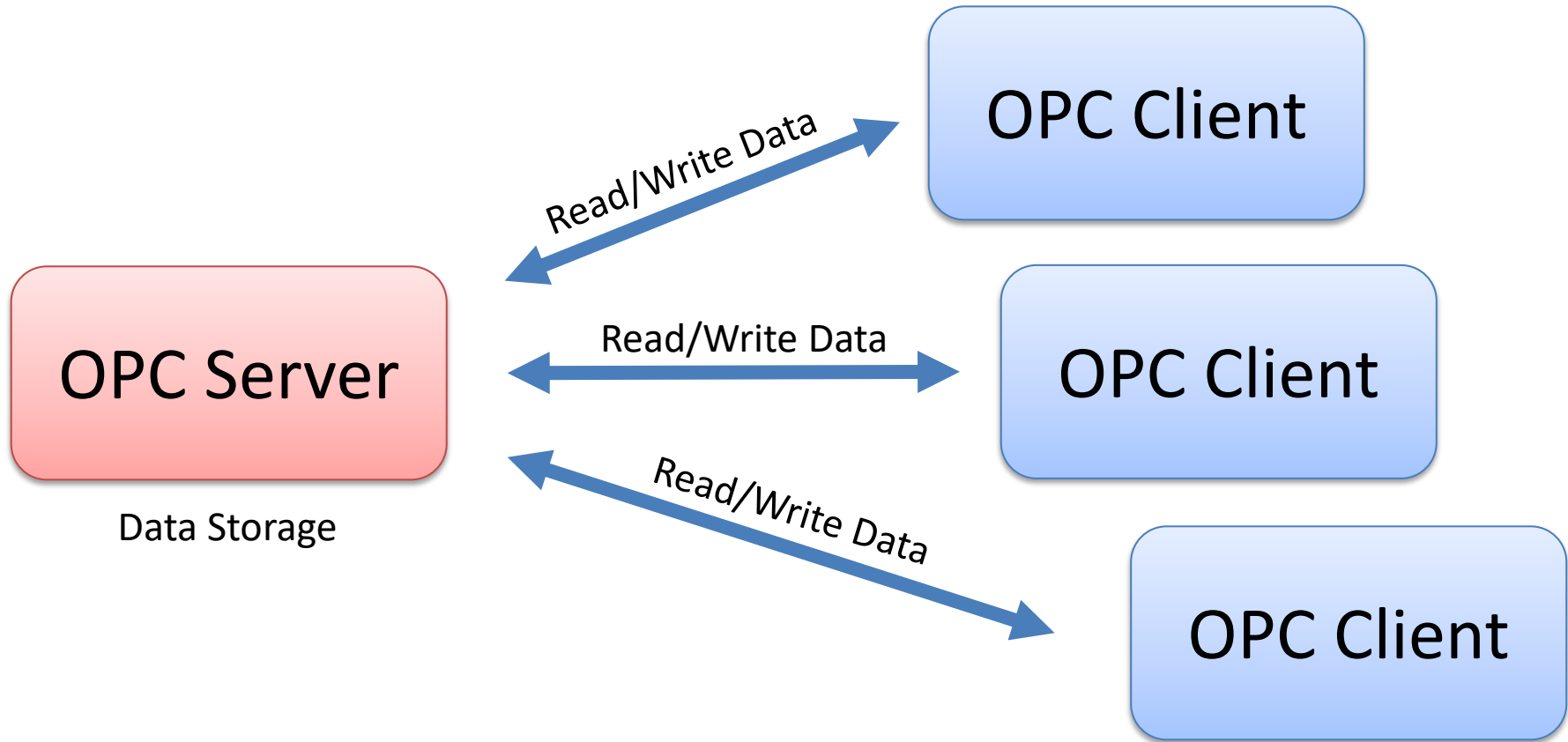


What is OPC?

What is OPC?

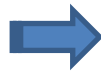
- OPC is a standard that defines the communication of data between devices from different manufactures
- OPC requires an **OPC Server** that communicates with one or more **OPC Clients**
- OPC allows “plug-and-play”, gives benefits as reduces installation time and the opportunity to choose products from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

OPC Server and Clients



OPC Specifications

“Classic” OPC



“Next Generation” OPC

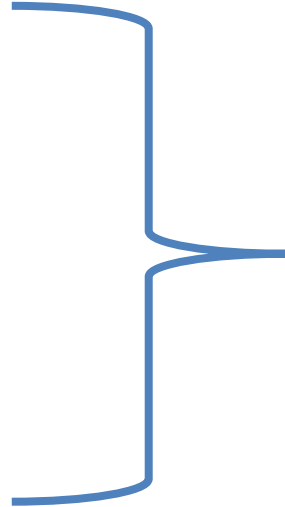
OPC DA

OPC HDA

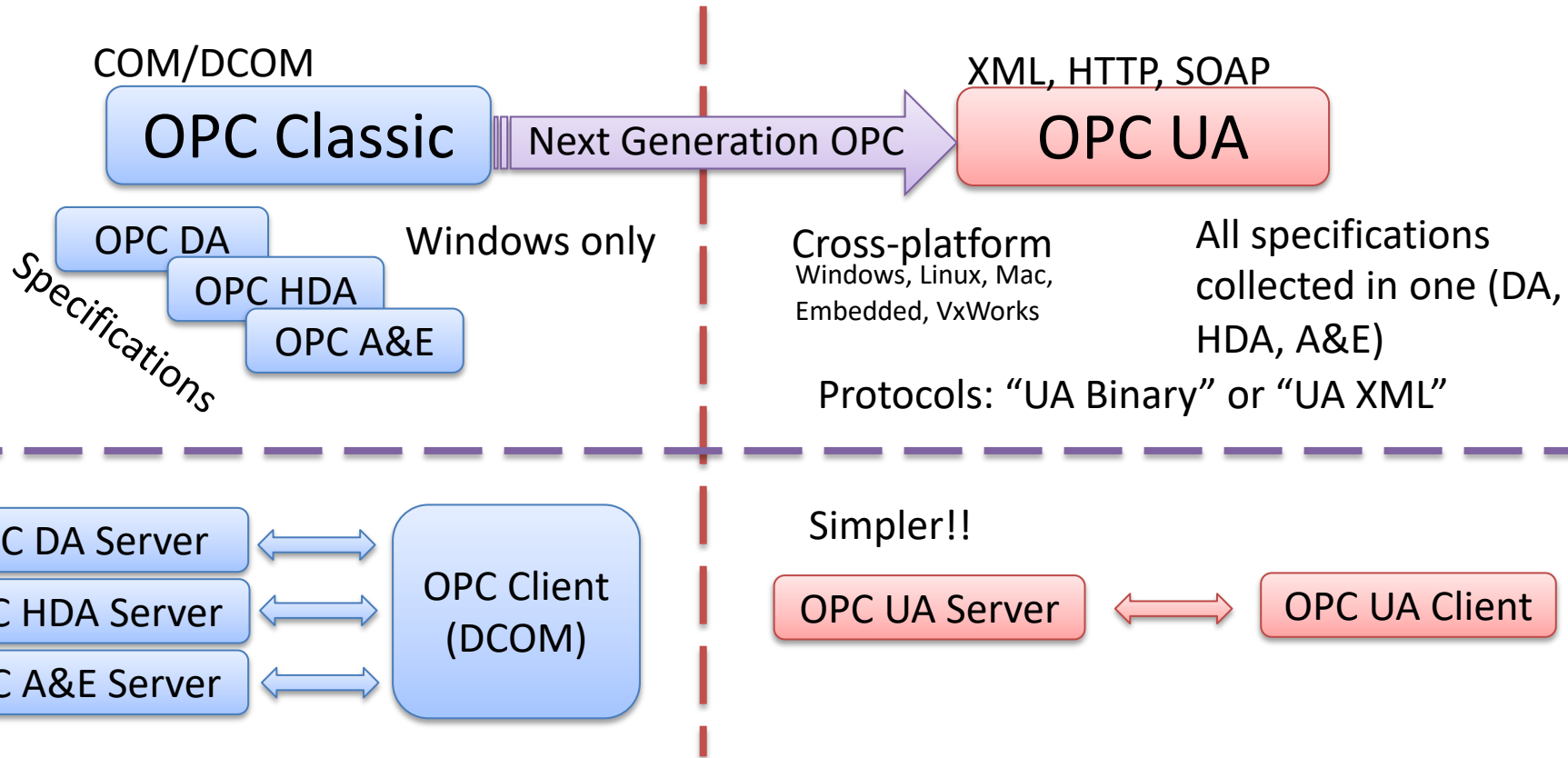
OPC A&E

... (Many others)

OPC UA



Next Generation OPC

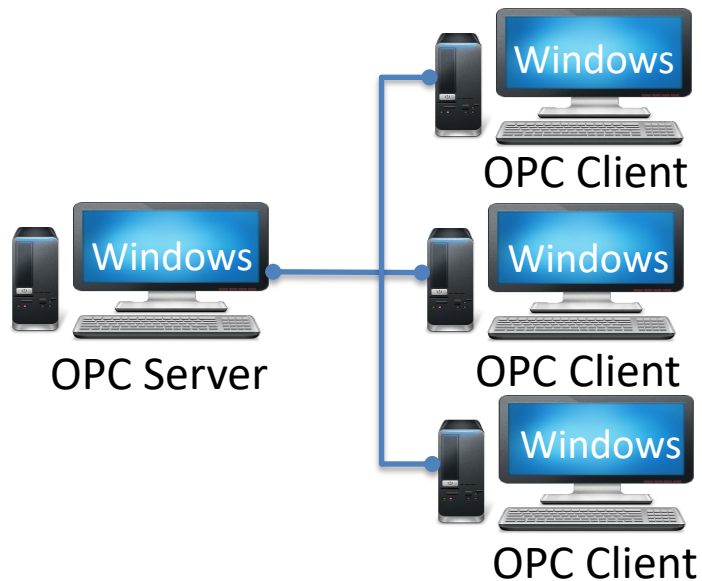


OPC UA

- 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

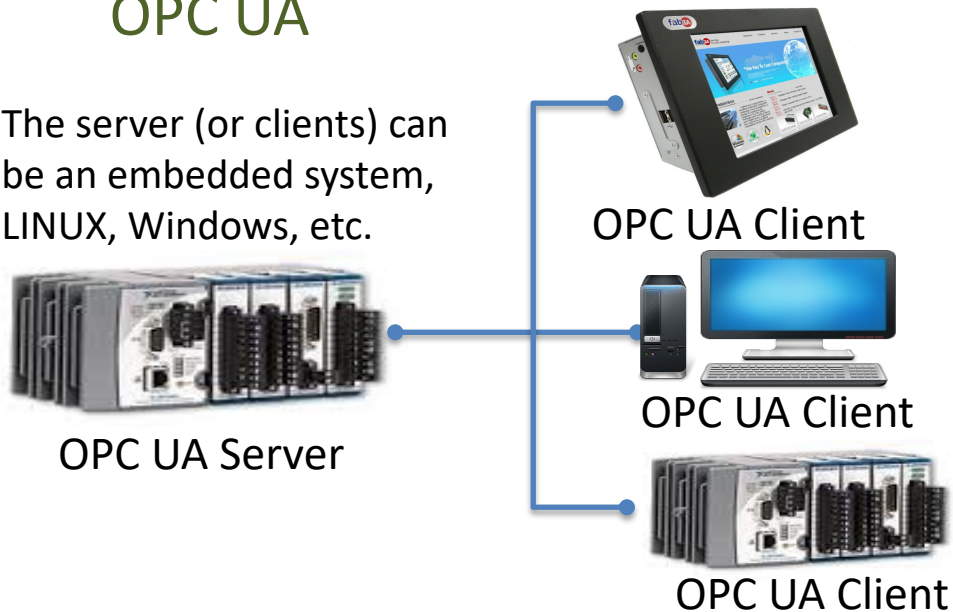
Classic OPC vs. OPC UA

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 with MATLAB

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

Hans-Petter Halvorsen

[Table of Contents](#)

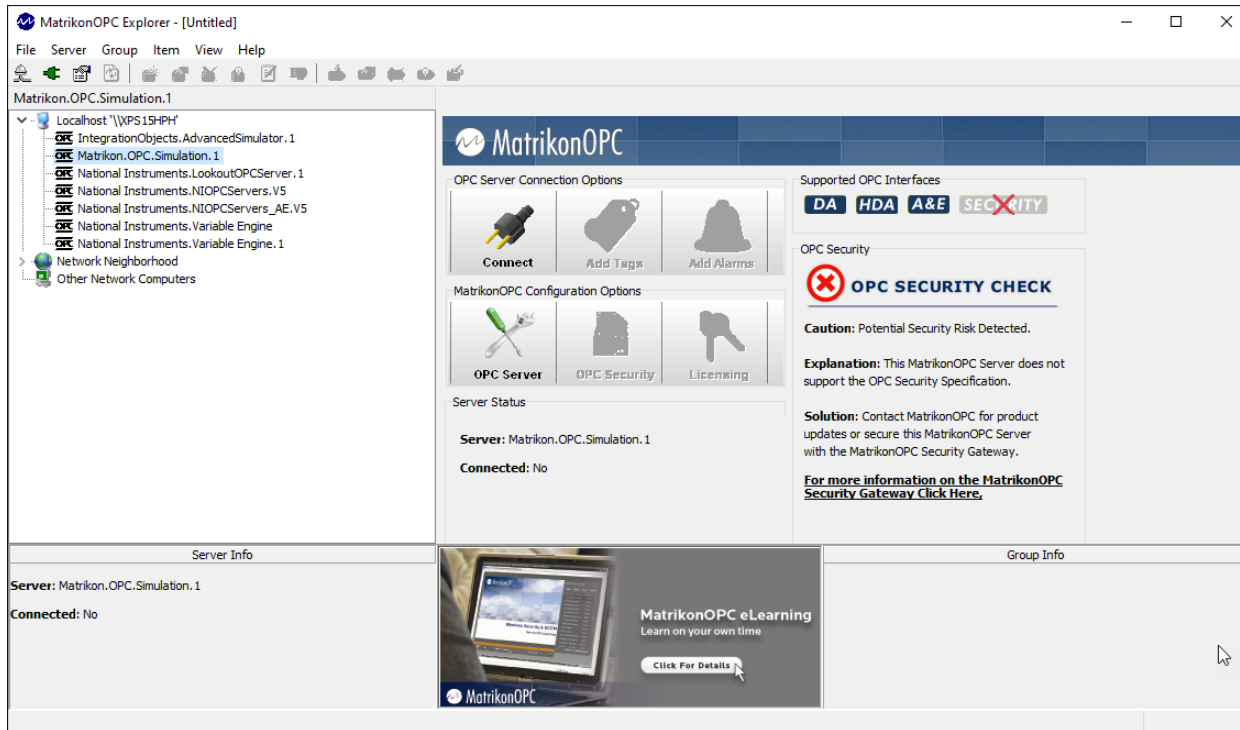
Industrial Communication Toolbox

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



OPC DA Examples

MatrikonOPC Simulation Server



MatrikonOPC
Simulation Server
is free and can be
used for testing
OPC DA

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



OPC DA - Example 1

Basic Example that reads a single value from an OPC Server

OPC DA - Example 1

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

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

%Add Tags
ItmList = {'Random.Real4'};
itm = additem(grp, ItmList);

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

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

This simple Example reads only one value from the OPC DA Server

OPC DA - Example 1b

% 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);
```

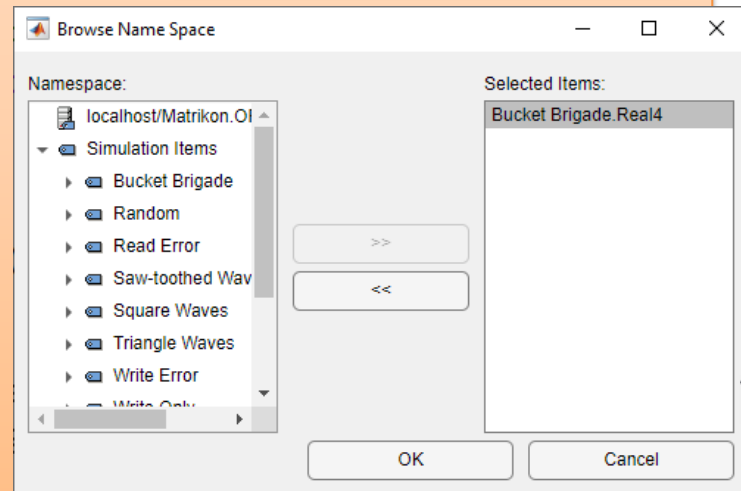
This Example makes it possible to select one or more OPC Tag from a pop-up window

% Retrieve Data

```
data = read(grp);  
opcdata = data.Value
```

%Clean Up

```
disconnect(da)  
delete(da)
```



OPC DA - Example 1c

```
% 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)
```

```
data.ItemID
```

```
data.Value
```

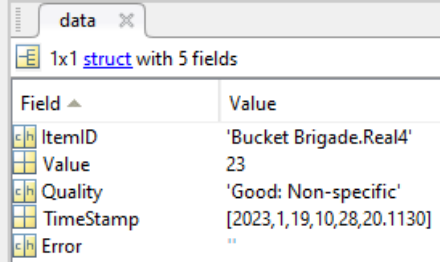
```
data.TimeStamp
```

```
data.Quality
```

```
%Clean Up
```

```
disconnect(da)
```

```
delete(da)
```



Field ▲	Value
ItemID	'Bucket Brigade.Real4'
Value	23
Quality	'Good: Non-specific'
TimeStamp	[2023, 1, 19, 10, 28, 20.1130]
Error	''

You can also get other information than the Value itself, like TimeStamp, Quality, etc.

OPC DA - Example 1d

```
% Connect to OPC Server
```

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

```
% Create Group
```

```
grp = addgroup(da, 'DemoGroup');
```

```
%Add Tags
```

```
ItmList = {'Bucket Brigade.Real4', 'Bucket Brigade.Real8'};  
itm = additem(grp, ItmList);
```

```
% Retrieve Data
```

```
data = read(grp);  
opcdata1 = data(1).Value  
opcdata2 = data(2).Value
```

Here we get data from
2 different OPC Tags

```
%Clean Up
```

```
disconnect(da)  
delete(da)
```

Fields	ItemID	Value	Quality	TimeStamp	Error
1	'Bucket Brigade.Real4'	23	'Good: Non-specific'	[2023,1,19,10,28,20.1130]	[]
2	'Bucket Brigade.Real8'	19	'Good: Non-specific'	[2023,1,19,13,51,48.2450]	[]
3					



OPC DA - Example 2

Basic Example that reads continuously from an OPC Server

OPC DA - Example 2

% Connect to OPC Server

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

% Create Group

```
grp = addgroup(da, 'DemoGroup');
```

%Add Tags

```
ItmList = {'Random.Real8'};  
itm = additem(grp, ItmList);
```

% Retrieve Data

N=10;

```
for i=1:N  
    data = read(grp);  
    opcdata(i) = data.Value;  
    pause(10)  
end
```

%Clean Up

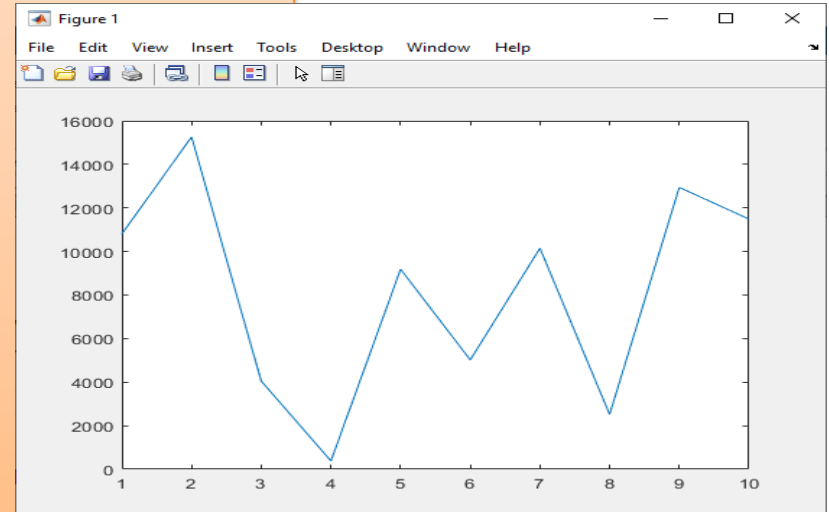
```
disconnect(da)  
delete(da)
```

%Present Data

```
plot(opcdata)
```

This simple Example reads values
from a OPC DA Server.

This Examples reads N values
using a For Loop





OPC DA - Example 3

Advanced Example that reads continuously from an OPC Server

Hans-Petter Halvorsen

[Table of Contents](#)

OPC DA - Example 3

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

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

%Add Tags
itmIDs = {'Random.Real8'};
itm = additem(grp, itmIDs);

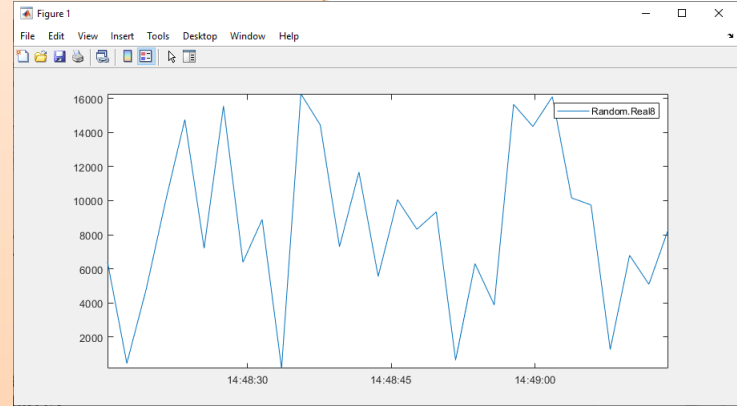
% Set Properties
logDuration = 60;
logRate = 2;
numRecords = ceil(logDuration./logRate);
grp.UpdateRate = logRate;
grp.RecordsToAcquire = numRecords;

% Acquire Data
start(grp);
wait(grp);

% Retrieve Data
[logIDs, logVal, logQual, logTime, logEvtTime] = getdata(grp, 'double');

% Plot Data
plot(logTime, logVal);
axis tight
datetick('x', 'keeplimits')
legend(logIDs);

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



This simple Example uses some of the more advanced features in the MATLAB Industrial Communication Toolbox.
No For/While Loop needed!



OPC UA Examples

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

[Digital Transformation](#)[OPC Products](#)[Services](#)[Training](#)[Company](#)[Partners](#)[Downloads](#)[Contact Us](#)[Q](#)

Home / OPC Products / OPC UA / OPC UA Server Simulator

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

DownloadUser 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 Client OPC UA Client OPC UA Client

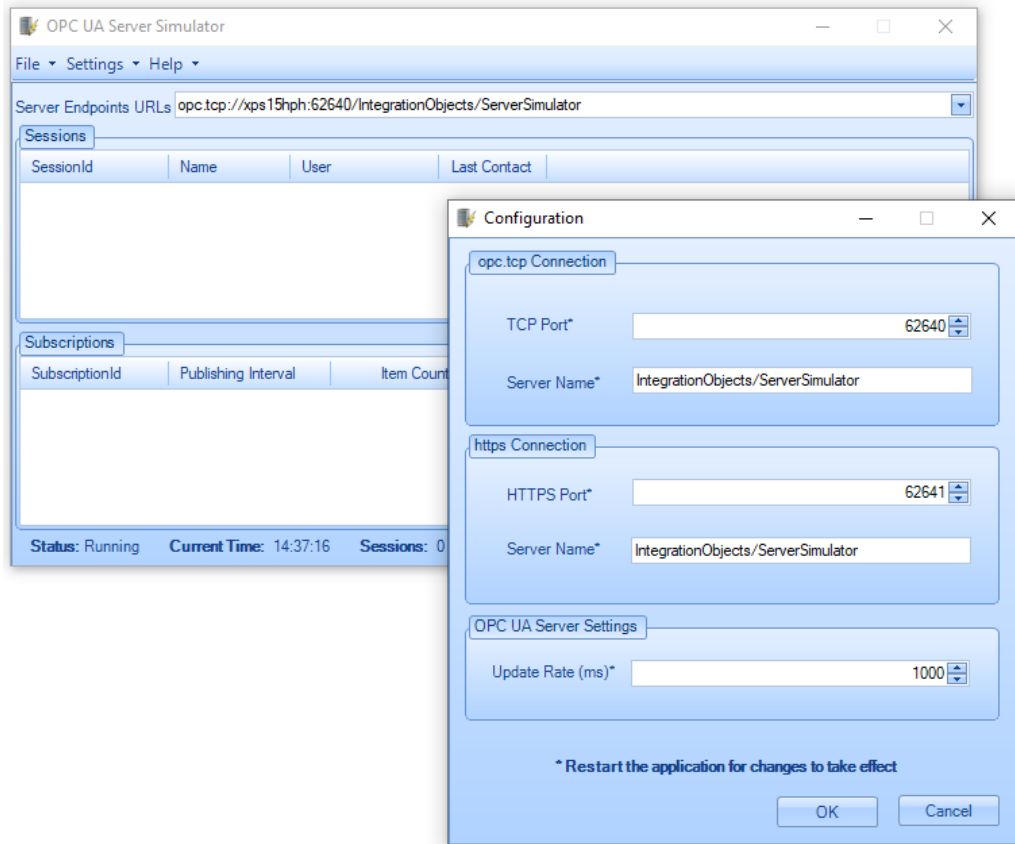
HTTP/UA TCP HTTP/UA TCP HTTP/UA TCP

OPC UA Server Simulator CSV CSV CSV

Privacy & Cookies Policy

<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

AutoSave Off AddressSpace.csv

File Home Insert Draw Page Layout Formulas Data Review View Add-Ins Help LOI Acro Tean

A1 Tag Name

	A	B	C	D	E	F
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						

AddressSpace

Ready Accessibility: Unavailable

AutoSave Off ValueSpace.csv

File Home Insert Draw Page Layout Formulas Data Review View Add-Ins Help LOI Acro Tean

A1 Tag11

	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									


ValueSpace

Ready Accessibility: Unavailable

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



Welcome Halvorsen Hans-Petter ? Ask Us a Question EN

Digital TransformationOPC ProductsServicesTrainingCompanyPartnersDownloadsContact Us

Home / OPC Products / OPC UA / OPC UA Client

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 Client

DownloadUser GuideQuick 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

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

OPC UA Client

The screenshot displays the 'Integration Objects' OPC UA Client application. The main window has a ribbon-style menu with tabs: Home, File, Session, Configuration, Help, and Certificate Manager. The 'Home' tab is active, showing icons for New, Open, Save, Save as, Connect, Disconnect, Settings, UA Settings, Help, About, Define, Remove, and Certificate Manager. The 'Sessions' pane on the left is empty. The 'Address Space' pane is also empty. The 'Forward' pane is empty. The 'Data View' pane is empty. The 'Connection Settings' dialog box is open, showing the following fields:

- Session Information: Session Name (Session0)
- Server Information: Endpoint Url (opc.tcp://xps15hph:62640/IntegrationObjects/), Discover button
- Transport Protocol: Opc.tcp (selected), Https
- Message Encoding: Binary (selected), Xml
- Security Mode: None (selected), Sign, Sign_Encrypt
- Security Policy: None (selected), Basic128RSA15, Basic256, Basic256Sha256
- User Authentication Mode: Anonymous (selected), UserName, Certificate
- Certificate (.pfx):
- Password:

The 'Apply' and 'Cancel' buttons are at the bottom of the dialog box. The main window also shows a 'Messages' pane at the bottom with the following data:

Message Type	Timestamp	Message
[Control]	2022-02-08 13:05:06	Disconnecting from session
[Control]	2022-02-08 13:03:09	Read operation of the variable
[Control]	2022-02-08 13:01:03	A session "Session0" with the

3 Messages

OPC UA Client

The screenshot shows the 'Integration Objects OPC UA Client' application window. The interface includes a top toolbar with icons for File (New, Open, Save, Save as), Session (Connect, Disconnect), Configuration (Settings, UA Settings), Help (Help, About), Default Configuration (Define, Remove), and Certificate (Certificate Manager). Below the toolbar are three tabs: Sessions, Data View, History View, and Event View. The Sessions tab on the left shows a tree view with 'Session0'. The Data View tab is active, displaying a large table with columns: Display Name, Node Id, Value, Data Type, Server Timestamp, Source Timestamp, Status Code, Subscription, and Session. The Address Space pane on the left shows a tree view under 'Forward' with 'Real Time Data' containing tags Tag1 through Tag7. A context menu is open over Tag7, showing options: References and Attributes, Read (highlighted), Write, History Update, and Monitor. The bottom status bar shows a message log with two entries: a successful read operation for Tag7 and a successful session creation message.

Integration Objects OPC UA Client

Home

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

Sessions

Session0

Address Space

Forward

Real Time Data

- Tag1
- Tag2
- Tag3
- Tag4
- Tag5
- Tag6
- Tag7
- Tag
- Tag

References and Attributes

Read

Write

History Update

Monitor

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

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

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



OPC UA - Example 4

OPC UA Server Simulator – Read Data

Hans-Petter Halvorsen

[Table of Contents](#)

MATLAB OPC UA - Functions

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 Server Simulator

The image displays two software interfaces for OPC UA. The left window is the 'OPC UA Server Simulator', and the right window is the 'Integration Objects' OPC UA Client.

OPC UA Server Simulator

Server Endpoints URLs: `opc.tcp://xps15ph:62640/IntegrationObjects/ServerSimulator`

Sessions

SessionId	Name	User	Last Contact
Session0	Anonymous	ns=3;i=1886064373	13:00:18

Subscriptions

SubscriptionId	Publishing Interval	Item Count	Seq No
	1000	1	2

Status: Running Current Time: 13:00:22 Sessions: 1 Subscriptions: 1

Integration Objects' OPC UA Client

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

Sessions

- Sessions
 - Session0
 - Subscription0

Data View | History View | Event View

Display Name	Node Id	Value	Data Type	Server Timestamp	Source Timestamp	Status Code	Subscription	Session
Tag7	ns=2;s=Tag7	22	Double	20-01-2023 1...	20-01-2...	Good	Subscription0	Session0

Address Space

Forward

- Tag1
- Tag2
- Tag3
- Tag4
- Tag5
- Tag6
- Tag7
- Tag8
- Tag9

Edit Numeric Value

22,0000000000000000

OK Cancel

Message

Message Type	Timestamp	Message
[Control]	2023-01-20 12:59:34	Create Monitored Item [ns=2;s=Tag7] succeeded.
[Control]	2023-01-20 12:59:33	A new subscription with the following properties: [Display Name: Subscription0, Publishing Interval: 1000, Keep Alive Count: 10, Lifetime Count: 1000, Max Notifications Per Publish: 0, Priority: 255, PublishingEnabled: True] was successfully created.

4 Messages

OPC UA - Example 4

```
clear, clc
```

Read Data

```
uaClient = opcua('localhost', 62640)
```

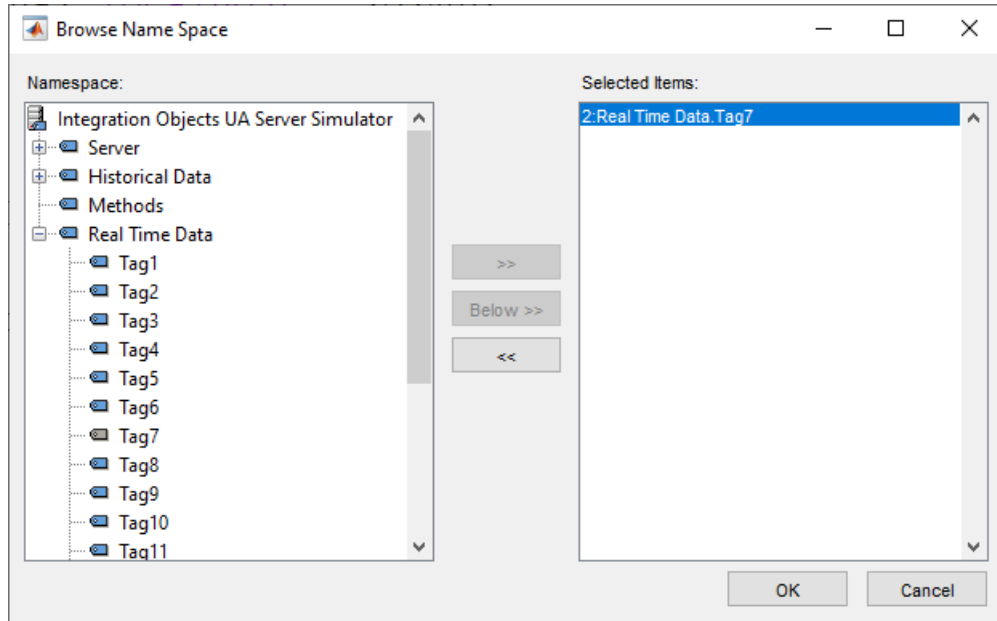
```
connect(uaClient)
```

```
serverNodes = browseNamespace(uaClient)
```

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

```
disconnect(uaClient);
```

OPC UA Server Simulator



Command Window

value =

22

timestamp =

datetime

20-Jan-2023 13:01:44

quality =

OPC UA Quality ID:

Good

OPC UA - Example 4b

```
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 - Example 5

OPC UA Server Simulator – Write Data

Hans-Petter Halvorsen

[Table of Contents](#)

MATLAB OPC UA - Functions

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 - Example 5

```
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);
```

Write Data



OPC UA Server Simulator

The image displays two software interfaces side-by-side. The left interface is the 'OPC UA Server Simulator' and the right is the 'Integration Objects' OPC UA Client'.

OPC UA Server Simulator Interface:

- Server Endpoints URLs:** `opc.tcp://xps15hph:62640/IntegrationObjects`
- Sessions Table:**

SessionId	Name	User
Session0	Anonymous	ns=3;i=1886064373

- Subscriptions Table:**

SubscriptionId	Publishing Interval	Item Count
	1000	1

- Status:** Running **Current Time:** 13:00:22 **Sessions:** 1

Integration Objects' OPC UA Client Interface:

- Home** tab with icons for New, Open, Save, Save As, Connect, Disconnect, Settings, UA Settings, Help, About, Define, Remove, and Certificate Manager.
- Sessions View:** Shows a tree with 'Session0' and its 'Subscription0'.
- Address Space:** A list of tags (Tag1 through Tag10) with 'Tag7' selected.
- Data View Table:**

Display Name	Node Id	Value	Data Type	Server Timestamp	Source Timestamp	Status Code	Subscription	Session
Tag7	ns=2;s=Tag7	21.7	Double	20-01-2023 1...	20-01-2...	Good	Subscription0	Session0

- Message Log:**

Message Type	Timestamp	Message
[Control]	2023-01-20 12:59:34	Create Monitored Item [ns=2;s=Tag7] succeeded.
[Control]	2023-01-20 12:59:33	A new subscription with the following properties: [Display Name: Subscription0, Publishing Interval: 1000, Keep Alive Count: 10, Lifetime Count: 1000, Max Notifications Per Publish: 0, Priority: 255, PublishingEnabled: True] was successfully created.

- 4 Messages**

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

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

