EH2750 Exercise 5: OPC servers and clients

By Nicholas Honeth (nicholash@ics.kth.se)

Introduction

This exercise aims to familiarise you with using OPC for reading process data and issuing commands. For this purposes of getting to know the system you will be using the Matrikon OPC simulation server and client on your individual workstations.

> **Note:** The IEC 60870-5-10x OPC servers used with the ICS village look exactly the same from a client point of view (with the actual tags from the RTUs of course).

There is a brief evaluation of this exercise which is to be answered in a TXT file and emailed to the exercise leader. Open text editor now and paste in the questions so that you can fill them in as we go through the work. Keep your answers short please ☺

Objectives

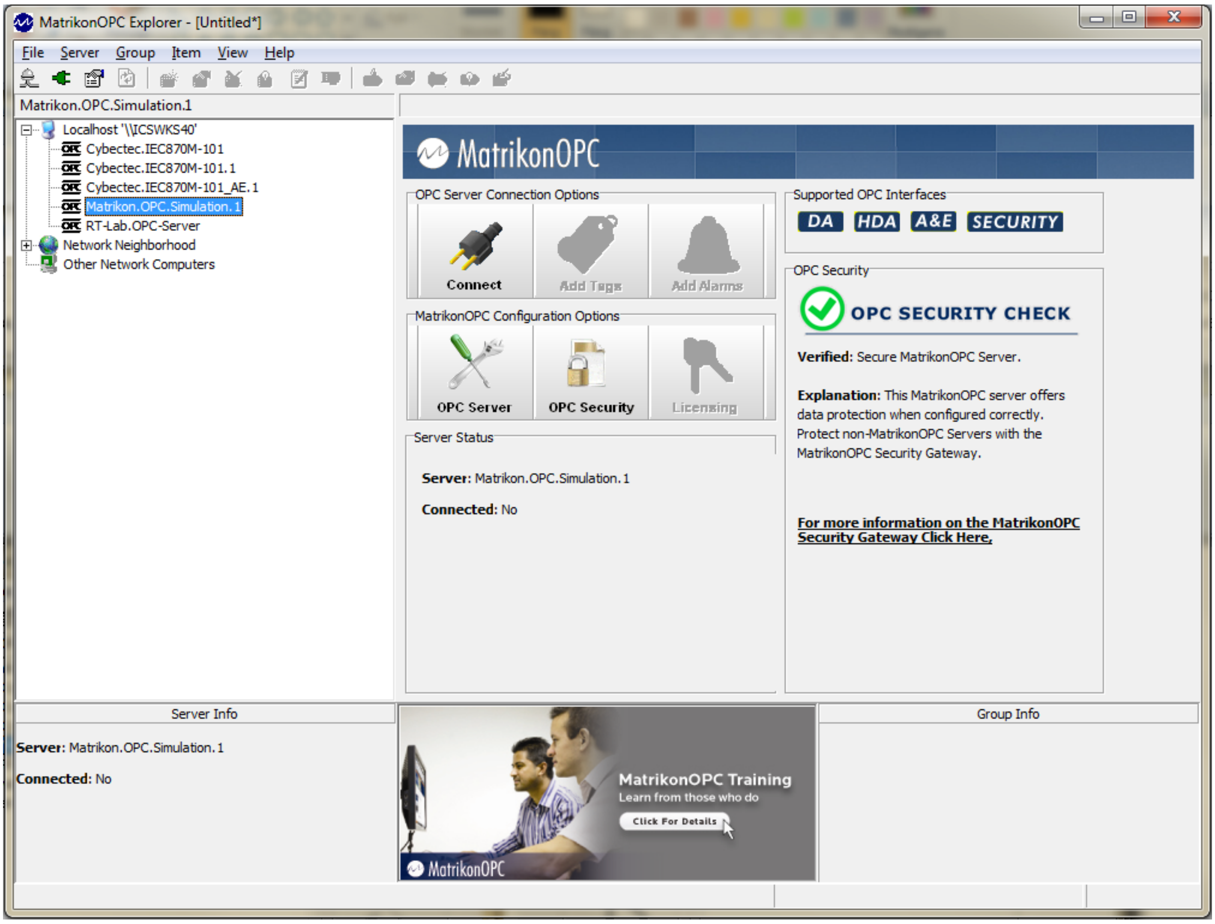
The objectives of exercise 5 are as follows:

- Understand the difference between OPC server and client
- Be able to start-up and connect the client to the server
- Understand the process of adding and browsing tags from the server
- Understand some of the important data types
- Be able to find and read data from specific tags
- Be able to find and write data to specific tags

Step 1: The OPC client

Start up the Matrikon OPC server explorer (which is an OPC client) from the start menu.

After you have started up the client tool (see diagram below) you will be presented with a frame containing servers you can connect to on your network on the left and a second (most likely empty) frame showing some alternatives (depending on what you select in the left frame).



Server: Matrikon.OPC.Simulation.1

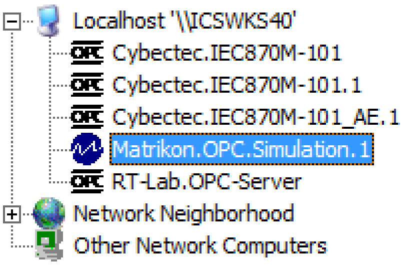
Connected: No

Verified: Secure MatrikonOPC Server.

Explanation: This MatrikonOPC server offers data protection when configured correctly. Protect non-MatrikonOPC Servers with the MatrikonOPC Security Gateway.

For more information on the MatrikonOPC Security Gateway Click Here.

The list of available hosts and OPC servers will look something like this:

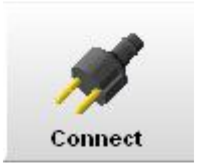


As you can see there are several servers accessible on “ICSWKS40”, some are the IEC 60870-5-10x servers, there is even the “RT-Lab.OPC.Server” for interfacing with our real-time simulator. You hosts will probably only have the “Matrikon.OPC.Simulation” servers available. This is what we will use in this exercise.

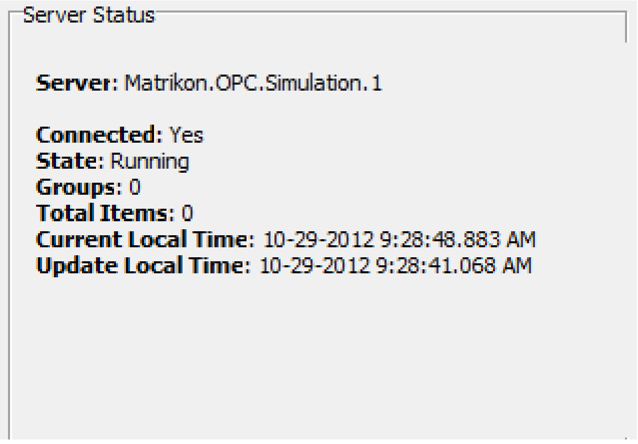
Step 2: Connecting to the server

Select the “Matrikon.OPC.Simulation” server from the list in the left frame.

Then connect to it using:



You will see that the server status frame will be updated like this:



Server: Matrikon.OPC.Simulation.1

Connected: Yes
State: Running
Groups: 0
Total Items: 0
Current Local Time: 10-29-2012 9:28:48.883 AM
Update Local Time: 10-29-2012 9:28:41.068 AM

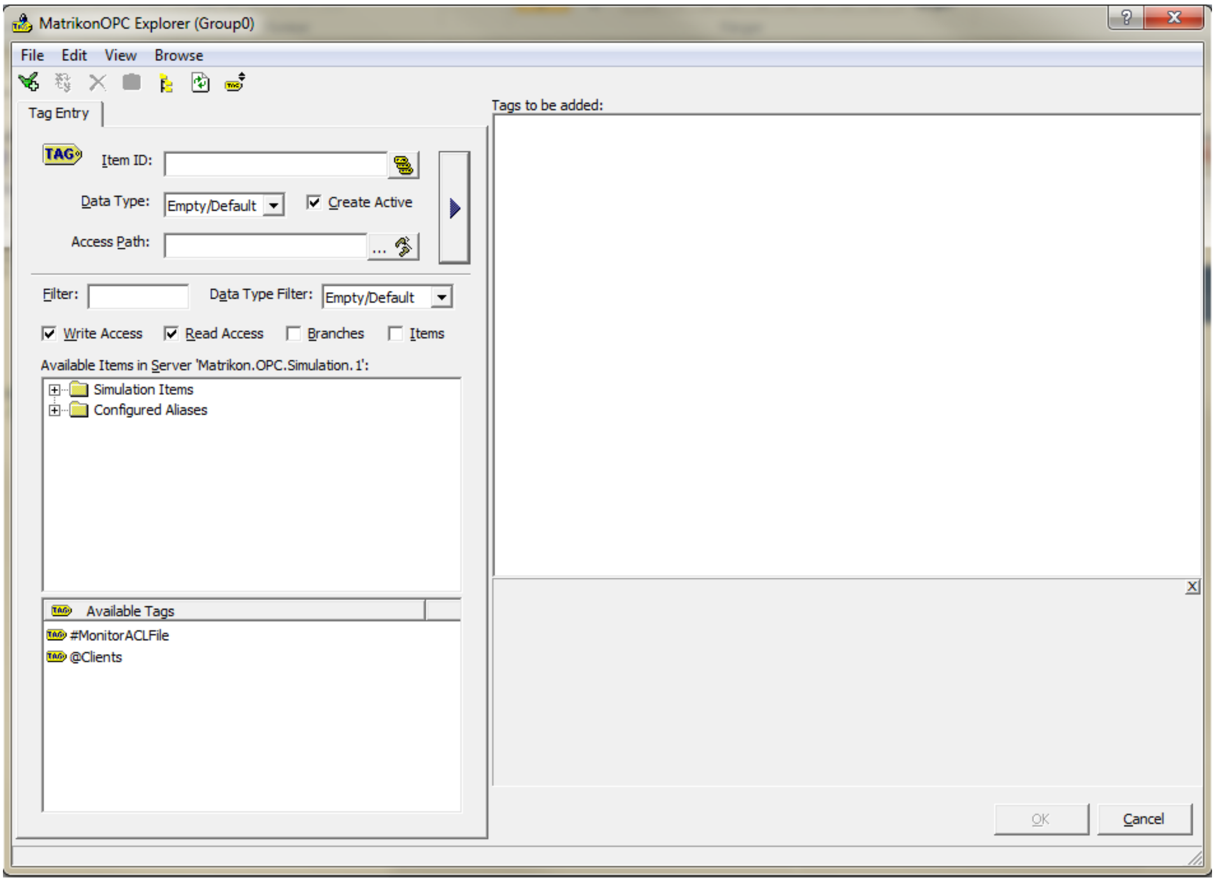
> **Note:** It is possible to connect to OPC servers on other hosts (computers). This configuration process for this is very tricky, that is why we use local simulation servers and have not connected you all directly to the IEC 60870-5-10x OPC servers for this exercise.

Step 3: Browsing the server

In the next step we want to see what data items (or **tags**) are accessible on the "Matrikon.OPC.Simulation" server.

To do this we select the Add Tags button.

When we do this a new dialog box will appear:



Item ID:

Data Type: Empty/Default ☑ Create Active

Access Path:

Filter: Data Type Filter: Empty/Default

☑ Write Access ☑ Read Access ☐ Branches ☐ Items

Available Items in Server 'Matrikon.OPC.Simulation.1':

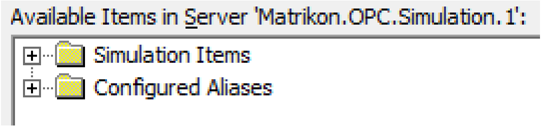
- Simulation Items
- Configured Aliases

Available Tags

- #MonitorACLFile
- @Clients

What has happened here is that the OPC explorer tool (OPC client) has created a new item/tag groups called "Group0" and is prompting us to add items/tags from the OPC server to this group.

The OPC server we are looking at has two groups, namely "Simulation Items" and "Configured Aliases":

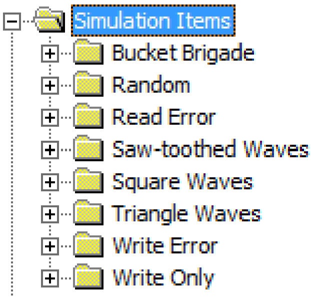


Available Items in Server 'Matrikon.OPC.Simulation.1':

- Simulation Items
- Configured Aliases

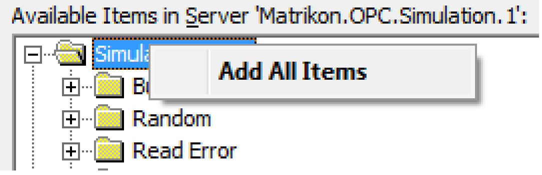
We will only focus on the simulation items for now.

If you expand the simulation items you will see that the OPC simulation server provides several different types of items to investigate:



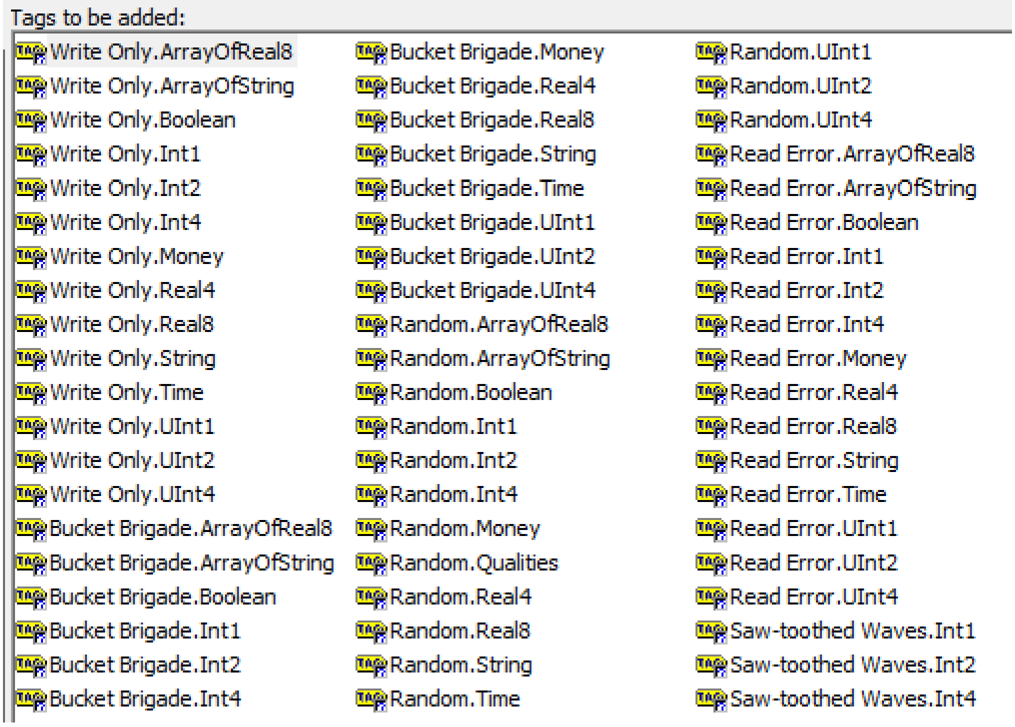
- Simulation Items
  - Bucket Brigade
  - Random
  - Read Error
  - Saw-toothed Waves
  - Square Waves
  - Triangle Waves
  - Write Error
  - Write Only

For the sake of this first investigation we can add all available items among the simulation items:



Right-click on “Simulation Items” and you will be able to select “Add All Items”.

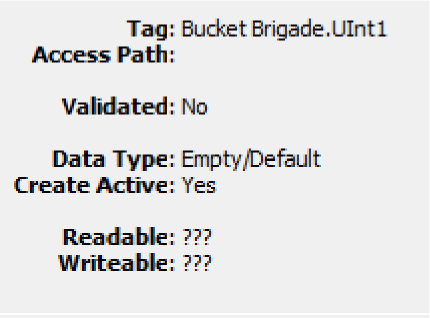
The “Tags to be added” frame on the right will then be filled with the items contained in the “Simulation Items” tree:



Tags to be added:

Write Only.ArrayOfReal8
Write Only.ArrayOfString
Write Only.Boolean
Write Only.Int1
Write Only.Int2
Write Only.Int4
Write Only.Money
Write Only.Real4
Write Only.Real8
Write Only.String
Write Only.Time
Write Only.UInt1
Write Only.UInt2
Write Only.UInt4
Bucket Brigade.ArrayOfReal8
Bucket Brigade.ArrayOfString
Bucket Brigade.Boolean
Bucket Brigade.Int1
Bucket Brigade.Int2
Bucket Brigade.Int4
Bucket Brigade.Money
Bucket Brigade.Real4
Bucket Brigade.Real8
Bucket Brigade.String
Bucket Brigade.Time
Bucket Brigade.UInt1
Bucket Brigade.UInt2
Bucket Brigade.UInt4
Random.ArrayOfReal8
Random.ArrayOfString
Random.Boolean
Random.Int1
Random.Int2
Random.Int4
Random.Money
Random.Qualities
Random.Real4
Random.Real8
Random.String
Random.Time
Random.UInt1
Random.UInt2
Random.UInt4
Read Error.ArrayOfReal8
Read Error.ArrayOfString
Read Error.Boolean
Read Error.Int1
Read Error.Int2
Read Error.Int4
Read Error.Money
Read Error.Real4
Read Error.Real8
Read Error.String
Read Error.Time
Read Error.UInt1
Read Error.UInt2
Read Error.UInt4
Saw-toothed Waves.Int1
Saw-toothed Waves.Int2
Saw-toothed Waves.Int4

When you select one of the tags in the list you will see that some information about that tag will be shown in the grey box below:



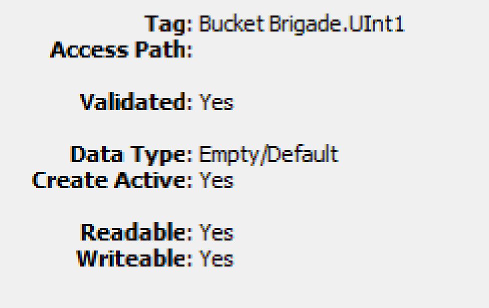
Tag: Bucket Brigade.UInt1
Access Path:

Validated: No

Data Type: Empty/Default
Create Active: Yes

Readable: ???
Writeable: ???

You will notice that some of the information is unknown. It can be found by clicking the “Validate” button. After this you should at least be able to see the read/write access to the tag:



Tag: Bucket Brigade.UInt1
Access Path:

Validated: Yes

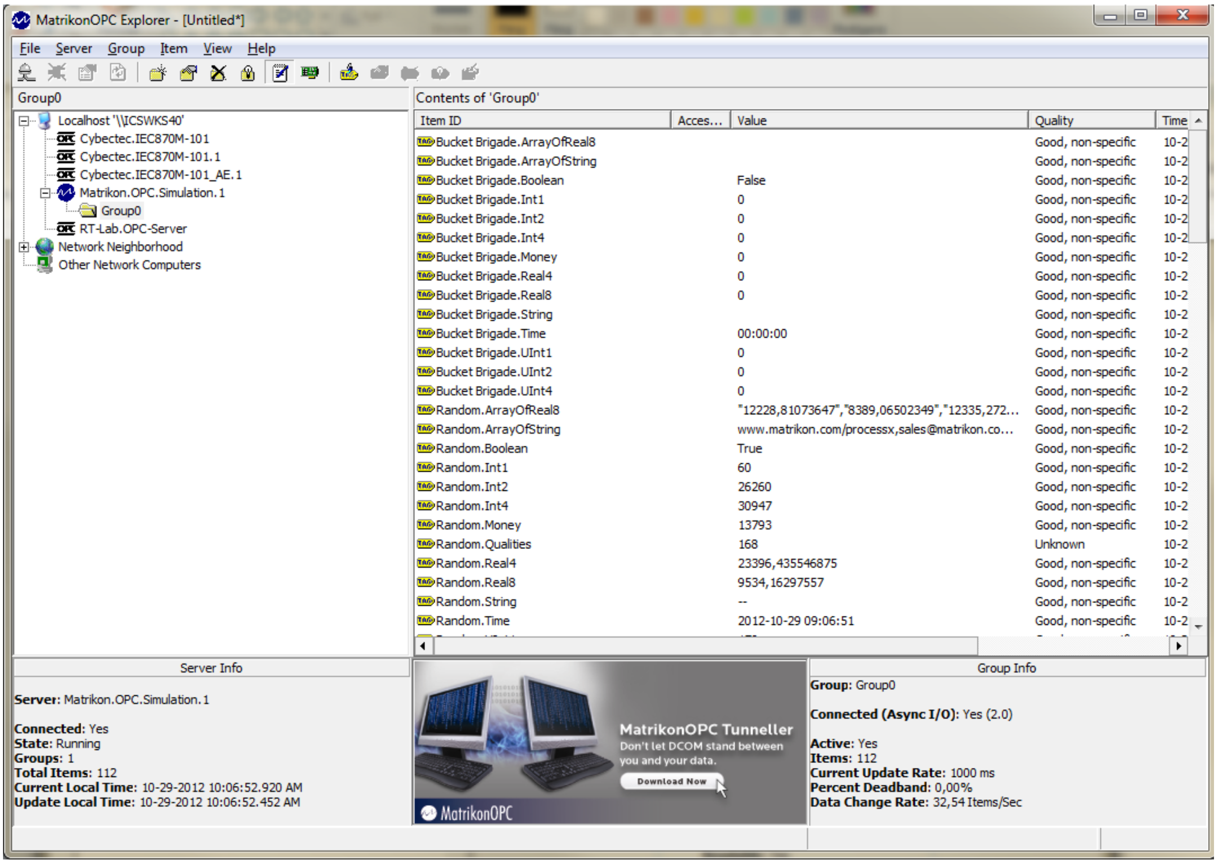
Data Type: Empty/Default
Create Active: Yes

Readable: Yes
Writeable: Yes

Now it is time to accept the addition of these tags so that we can have a look at the process data that is accessible through the OPC server. Click “Ok” to continue.

Step 4: Reading data from tags

Once the tags have been added you should be brought back to the main OPC explorer window:



Contents of 'Group0'

Item ID	Acces...	Value	Quality	Time
Bucket Brigade.ArrayOfReal8			Good, non-specific	10-2
Bucket Brigade.ArrayOfString			Good, non-specific	10-2
Bucket Brigade.Boolean		False	Good, non-specific	10-2
Bucket Brigade.Int1		0	Good, non-specific	10-2
Bucket Brigade.Int2		0	Good, non-specific	10-2
Bucket Brigade.Int4		0	Good, non-specific	10-2
Bucket Brigade.Money		0	Good, non-specific	10-2
Bucket Brigade.Real4		0	Good, non-specific	10-2
Bucket Brigade.Real8		0	Good, non-specific	10-2
Bucket Brigade.String			Good, non-specific	10-2
Bucket Brigade.Time		00:00:00	Good, non-specific	10-2
Bucket Brigade.UInt1		0	Good, non-specific	10-2
Bucket Brigade.UInt2		0	Good, non-specific	10-2
Bucket Brigade.UInt4		0	Good, non-specific	10-2
Random.ArrayOfReal8		"12228,81073647","8389,06502349","12335,272...	Good, non-specific	10-2
Random.ArrayOfString		www.matrikon.com/processx,sales@matrikon.co...	Good, non-specific	10-2
Random.Boolean		True	Good, non-specific	10-2
Random.Int1		60	Good, non-specific	10-2
Random.Int2		26260	Good, non-specific	10-2
Random.Int4		30947	Good, non-specific	10-2
Random.Money		13793	Good, non-specific	10-2
Random.Qualities		168	Unknown	10-2
Random.Real4		23396,435546875	Good, non-specific	10-2
Random.Real8		9534,16297557	Good, non-specific	10-2
Random.String		--	Good, non-specific	10-2
Random.Time		2012-10-29 09:06:51	Good, non-specific	10-2

Server Info

Server: Matrikon.OPC.Simulation.1

Connected: Yes
State: Running
Groups: 1
Total Items: 112
Current Local Time: 10-29-2012 10:06:52.920 AM
Update Local Time: 10-29-2012 10:06:52.452 AM

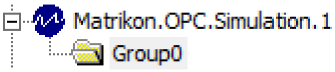
Group Info

Group: Group0

Connected (Async I/O): Yes (2.0)

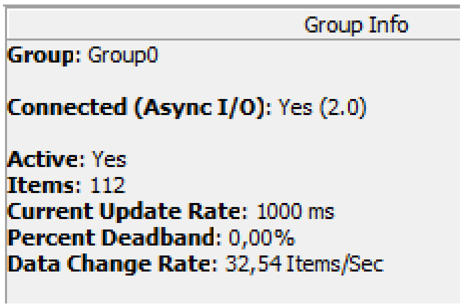
Active: Yes
Items: 112
Current Update Rate: 1000 ms
Percent Deadband: 0,00%
Data Change Rate: 32,54 Items/Sec

One thing that you should notice is that the automatically created “Group0” will now be shown under the “Matrikon.OPC.Simulation” server.



You will also notice that you are able to see the values changing in real-time as they are changing in the simulation.

You will be able to see some statistics on how the group is being read from the server in the lower-right area of the window:



Group Info

Group: Group0

Connected (Async I/O): Yes (2.0)

Active: Yes
Items: 112
Current Update Rate: 1000 ms
Percent Deadband: 0,00%
Data Change Rate: 32,54 Items/Sec

Also notice that there is some indication of the **data quality** this is an important piece of information as we will show in a short demonstration from the IEC 60870-5-100x OPC servers.

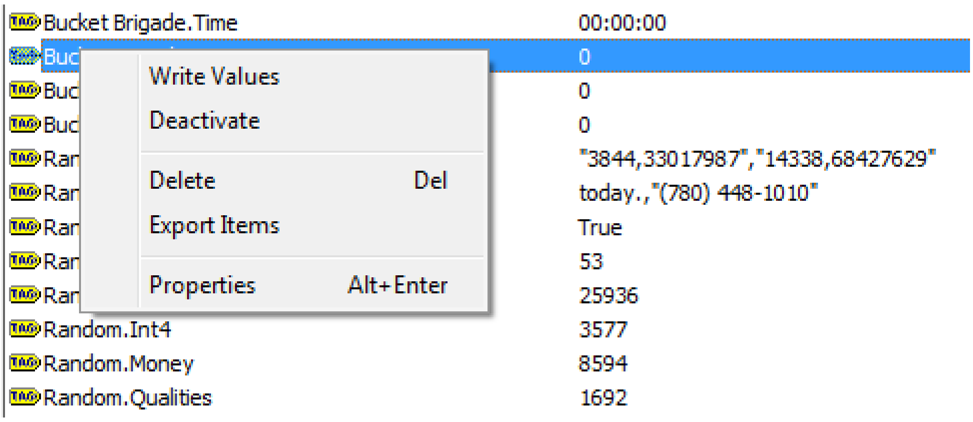
Step 5: Writing data to tags

So far we are able to see what is happening but we are not able to interact with it. We need to be able to write values back to the server. To do this we can select a *writeable* item from among the tags we have added to Group0.

Let's take "Bucket Brigade.UInt1" where "UInt" is an indication of the data type to use – in this case it is and *unsigned integer*.

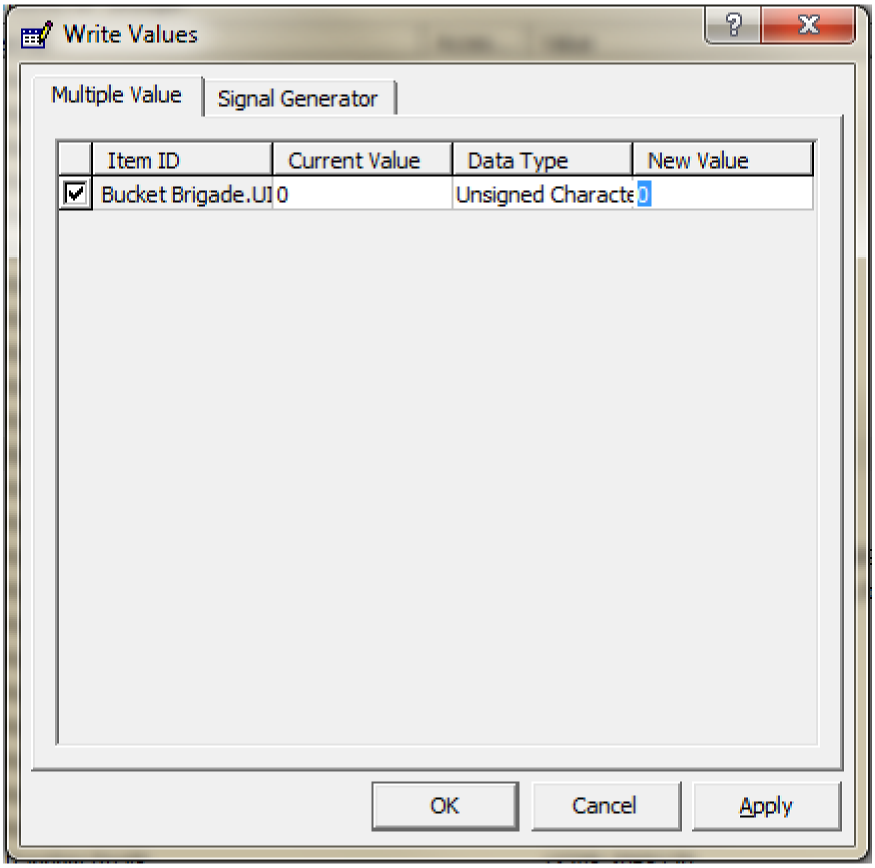
You'll notice that the values in the "Bucket Brigade" are writeable and are not changed by the simulation process so they are nice for testing whether we can write to the values.

Start by selecting "Bucket Brigade.UInt1" and right-clinking on it:



Bucket Brigade.Time	00:00:00
Buc	0
Buc	0
Buc	0
Ran	"3844,33017987","14338,68427629"
Ran	today.,"(780) 448-1010"
Ran	True
Ran	53
Ran	25936
Random.Int4	3577
Random.Money	8594
Random.Qualities	1692

Here we are interested in writing a value to "Bucket Brigade.UInt1" so we click on "Write Values". A new dialog box should appear:



	Item ID	Current Value	Data Type	New Value
☑	Bucket Brigade.UI	0	Unsigned Characte	0

There are two tabs available here. One can either write a particular value or use the *signal generator* for writing values to the tag. The signal generator is very useful for testing purposes.

Note: Notice here that the *DataType* shown in the field is of type “Unsigned Character” whereas from the name we suspect that the type should be an unsigned integer. OPC is fairly *loosely typed* meaning that it usually won’t give you errors when trying to write data of the wrong type to a tag. While this makes things work a little more easily it also means that there is a risk of strange results if you have been writing data types that are the same over parts of the range but not all of it – signed and unsigned types are typical examples of this.

We’ll just try writing a particular value.

First set the *Data Type* to the correct one – what is it?

Then set the *New Value* to some positive integer. Click “Ok” and watch the magic happen.

Try something similar with the *signal generator*, see whether you can count between 0-100.

Also, try writing with inconsistent data types in a few combinations to see what happens.

Summary

During this exercise we briefly went through the practical aspects of connecting to and interacting with an OPC server . The reason that it is important to understand this is because it is used as the middleware layer between your JACK applications and the IEC 60870-5-10x protocols for interfacing with RTUs 1, 2 and 3 connected to the village.

For the purposes of this exercise we have used a manual client – with a Human interface where you as a human can see values on the screen etc. When you use the ICS JACK platform in your projects you will need to configure the client connections on the OPCagent which will provide a transparent connection from your JACK applications to the RTUs (reading measurements, status and issuing commands). We will look at this during the next exercise.

Note: There is a lot to know about OPC – what we have looked at today are the absolute basics of interacting with the servers. If you are interested to learn some more about this come and ask us for some hints on what to read up on. OPC-UA is possible going to be included in IEC 61850 as an alternative to MMS – we have not confirmed this yet but it makes sense. Watch this space…

Evaluation

In order to evaluate you understanding of the contents of this exercise you are required to answer the following questions:

1. Explain the difference between the OPC server and client. What are the names of the ones used in the exercise?
2. What is your understanding of the meaning of items, tags and groups?
3. How did you get the 0-100 counting to work?
4. What happens when you try to write with inconsistent types? Why can this be a problem?

Your answers should be written in a TXT file and emailed to nicholash@ics.kth.se before you leave the lab.