

# Getting started PiiGAB M-Bus Explorer M-Bus OPC-server

[www.piigab.com](http://www.piigab.com)

## Contents

<b>1. DOCUMENT INFORMATION .....</b>	<b>3</b>
1.1 VERSIONS .....	3
<b>2. PRECONDITIONS .....</b>	<b>3</b>
<b>3. REQUIREMENTS.....</b>	<b>3</b>
3.1 OPTIONAL REQUIREMENTS .....	3
<b>4. CREATE A NEW OPC-SERVER PROJECT.....</b>	<b>4</b>
4.1 SET PROJECT AS ACTIVE PROJECT FOR THE OPC-SERVER .....	5
<b>5. READ-OUT DATA FROM THE M-BUS METER .....</b>	<b>5</b>
5.1 CREATE A CHANNEL (M-BUS MASTER) .....	5
5.2 IMPORT THE BROWSE TEMPLATE TO REPRESENT AN M-BUS METER.....	7
5.3 MONITOR THE TEMPLATE TO SEE WHAT IS IN THE M-BUS METER .....	9
<b>6. SELECT THOSE OPC-TAGS YOU WANT TO USE FOR YOUR SITE .....</b>	<b>12</b>
<b>7. CREATE A CONFIGURATION THAT SUITS YOUR SITE.....</b>	<b>13</b>
7.1 CREATE A DEVICE (METER) .....	13
<b>8. CONFIGURE NUMBER OF TELEGRAMS FOR A METER .....</b>	<b>17</b>
8.1 THEORETIC EXAMPLE .....	17
8.2 FIND TELEGRAMS IN METER .....	18
8.2.1 <i>Multi telegram meter</i> .....	19
8.2.2 <i>Single telegram meter</i> .....	19
8.3 SET NUMBER OF TELEGRAMS FOR A METER .....	20
<b>9. SAVE METER AS TEMPLATE (OPTIONAL) .....</b>	<b>21</b>
<b>10. TEST WITH AN OPC-CLIENT (OPTIONAL).....</b>	<b>23</b>
<b>11. APPENDIX.....</b>	<b>26</b>
11.1 CONTACTS .....	26

# 1. Document Information

This document will describe how to configure the PiiGAB Explorer for M-Bus communication with one M-Bus meter through an M-Bus master. For example a PiiGAB 810 or a PiiGAB 900 can be used as an M-Bus master.

If you see something that is not correct in this document, that misleads you or if you are missing something please contact us so we can improve this document continuously. See contact information at the end of the document.

## 1.1 Versions

Version	Modifies by	Detail
1.00.00	Stefan Eriksson	Initial version
1.00.01	Stefan Eriksson	Edited distributor contact information

## 2. Preconditions

Object	Detail/Other
One PiiGAB 810	You can use a PiiGAB 900 as well
One M-Bus meter with primary address set to 1	Supports EN13757
PiiGAB Explorer / M-Bus OPC-server	Version 2.00.00.000 or later
An established communication with the M-Bus meter, M-Bus master and PiiGAB M-Bus Setup wizard. See getting started PiiGAB 810.	
The PiiGAB 810 with IP-address set to 192.168.10.122, port 10001 and UDP as protocol	

## 3. Requirements

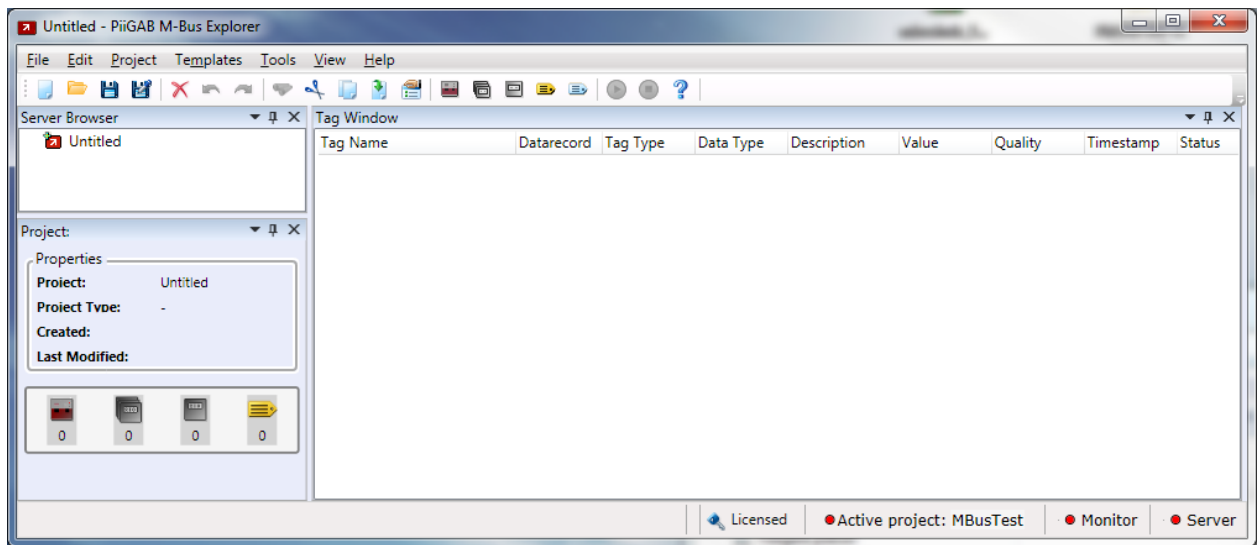
- PiiGAB Explorer

### 3.1 Optional requirements

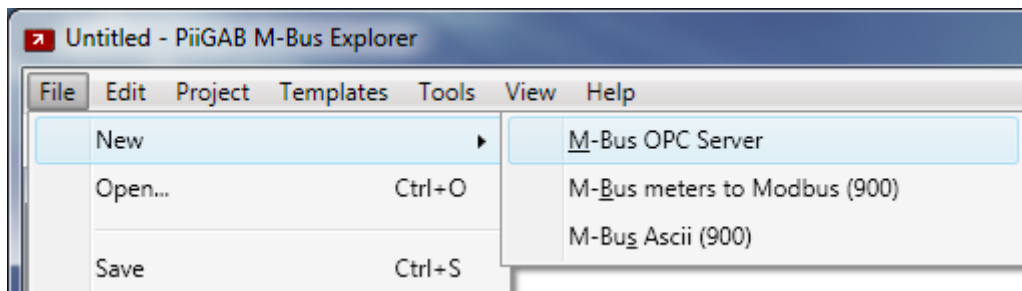
Softing's Demo client - <http://www.softing.com/en/index.html>

## 4. Create a new OPC-server project

1. Start PiiGAB Explorer

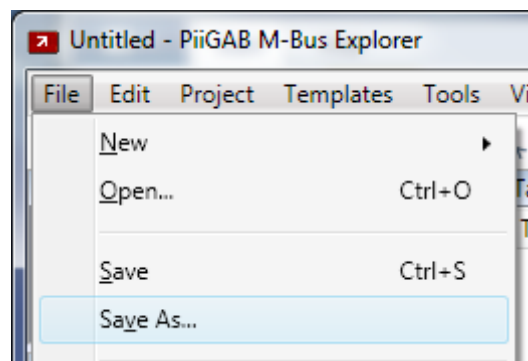


2. Go to *File* menu, select *New* and click on *M-Bus OPC Server*



An OPC-server project is created in PiiGAB Explorer

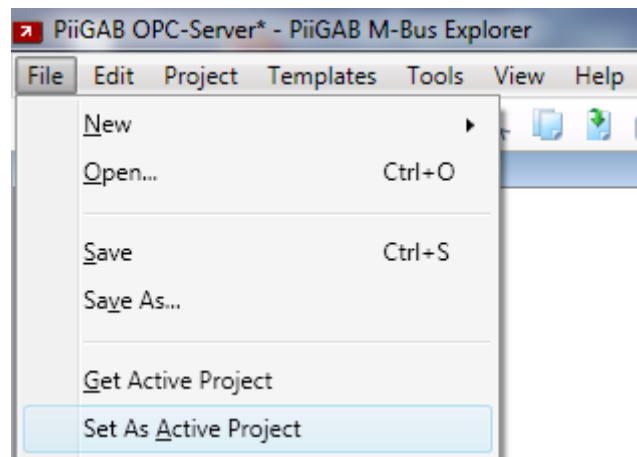
3. Go to *File* menu, click on *Save As...*



4. Save your project. This example saved the project as *PiiGAB OPC-server*.

## 4.1 Set project as active project for the OPC-server

1. Go to *File* menu and click on *Set As Active Project*



By setting the project as the active, the OPC-server will know which configuration file it will load when starting. You must also do this if you want to monitor the OPC-tags in PiiGAB Explorer.

2. Make sure your project is the active project in the bottom right corner of PiiGAB Explorer

● Active project: PiiGAB OPC-Server

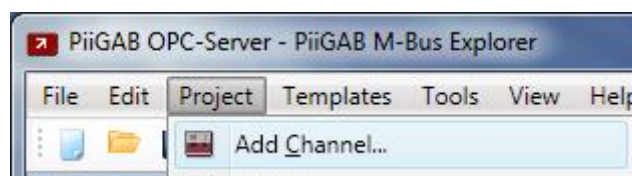
## 5. Read-out data from the M-Bus meter

To read the M-Bus meter you have to setup a channel and a preconfigured meter to see what's possible to acquire from it.

### 5.1 Create a channel (M-Bus master)

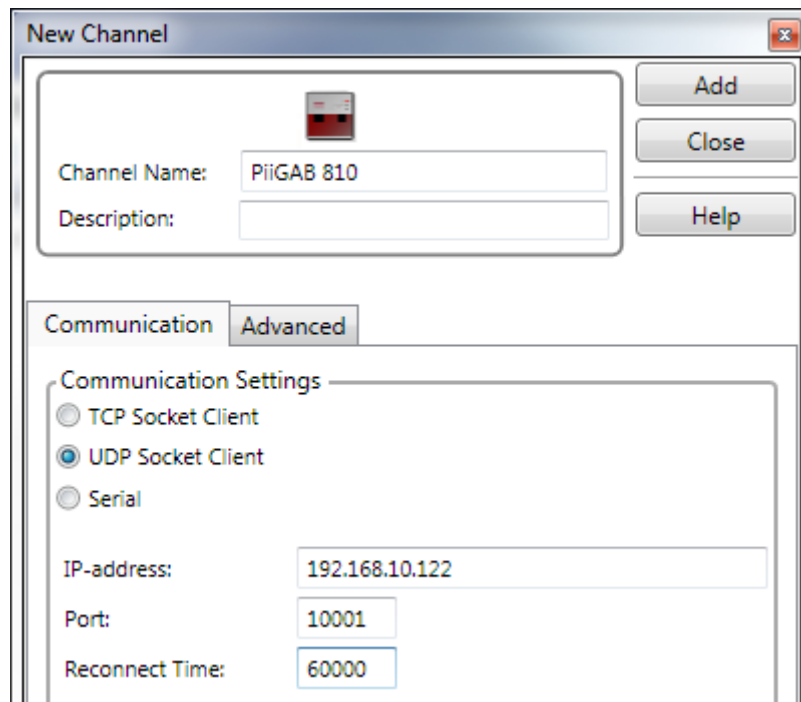
A channel in PiiGAB Explorer represents an M-Bus master. The channel needs the M-Bus master's communication parameters.

1. Go to *Project* menu and click on *Add Channel...*



You will see a window to configure the channel.

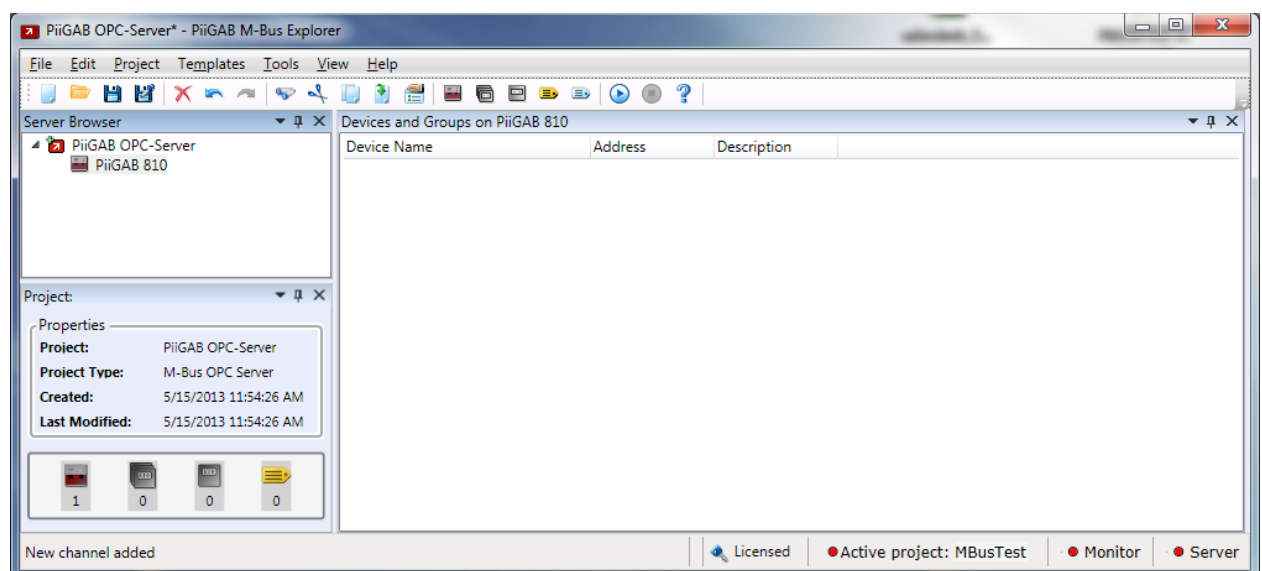
2. Configure the channel as specified in the picture below



**Note:**

The channel's settings are usually the same as used in the PiiGAB M-Bus Setup Wizard. Your gateway may have another IP-address; change the configuration for your setup. If your gateway communicates serial, choose serial settings instead.

3. Press *Add* to create the channel and add it in into the project's tree view



The tree view now displays the created channel.

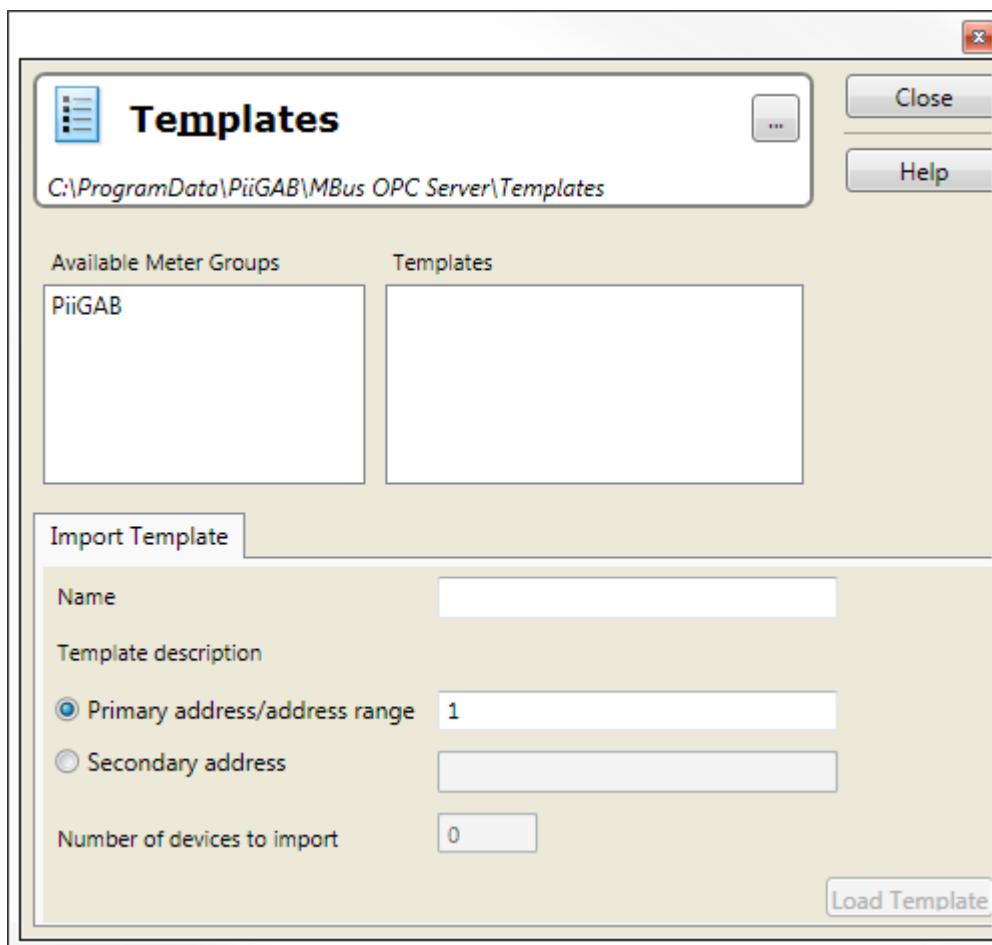
## 5.2 Import the Browse template to represent an M-Bus meter

Use the *Browse* template in PiiGAB Explorer to read-out the M-Bus meter and show its content.

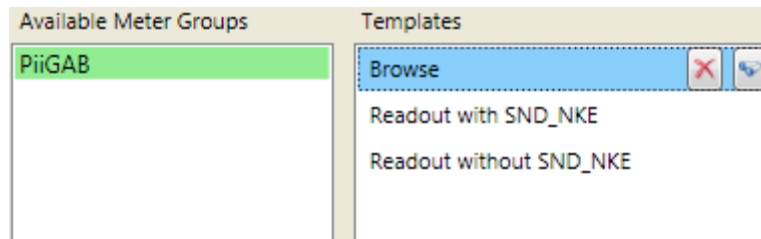
1. Go to *Template* menu and click on *Import Template...*



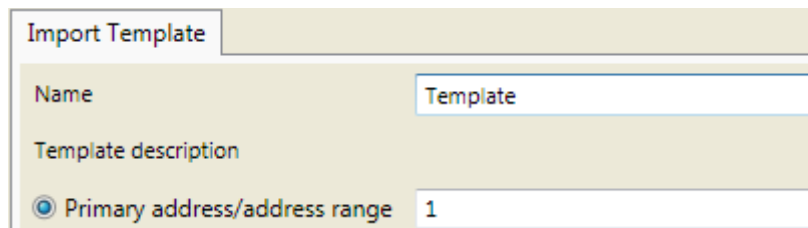
You will see a window where you can select a template for your channel.



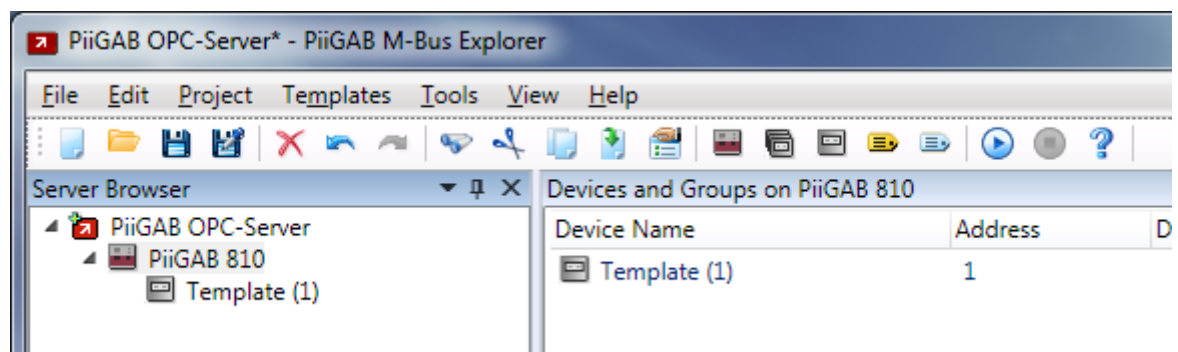
2. Click on *PiiGAB* in the left box (*Available Meter Groups*)
3. Click on *Browse* in the right box (*Templates*)



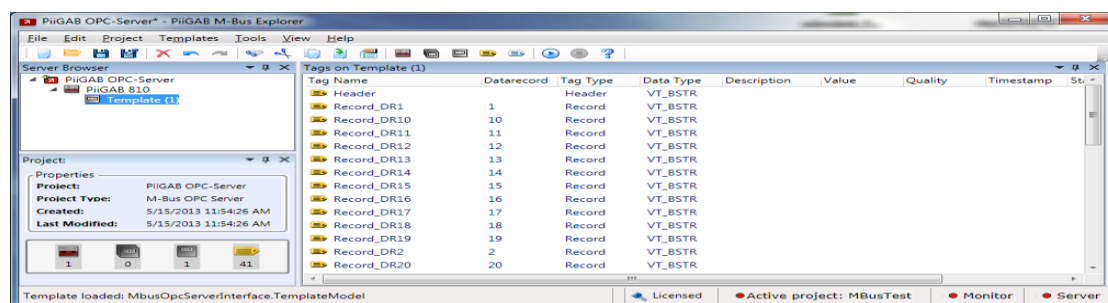
4. Configure the template as the picture below shows:



5. Press the *Load Template* button
6. Close the window with the *Close* button
7. Click the *Template (1)* meter in the tree view



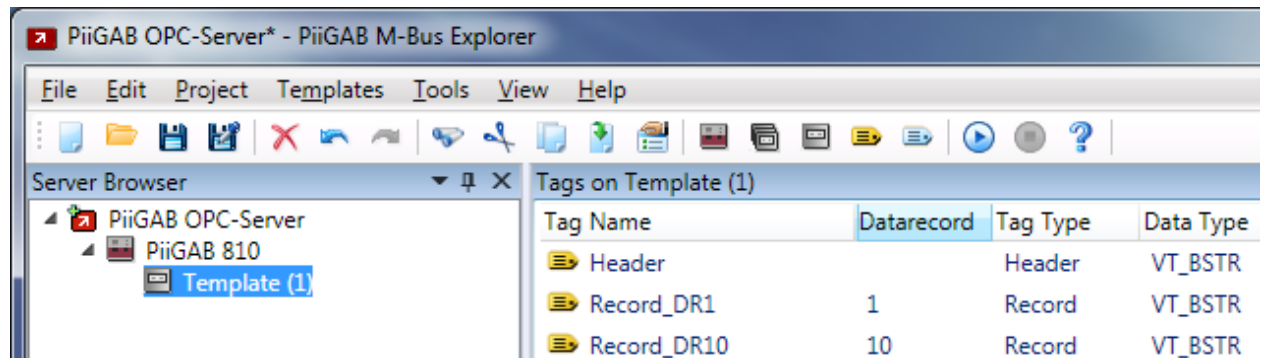
This will show a pre-configured meter which assumes that the M-Bus meter contains 40 objects. These 40 objects will be represented as 40 OPC-tags in the OPC-server.






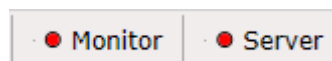
## 5.3 Monitor the template to see what is in the M-Bus meter

1. In the *Template (1)* meter click the *Datarecord* column



This will sort all objects in numerical order. You can also sort on tag name or any of the other columns in the table.

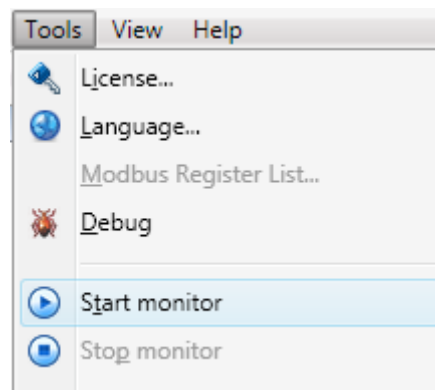
2. Save the project by pressing the  button in the top left corner
3. Make sure the *Server* monitor is steady red in the bottom right corner



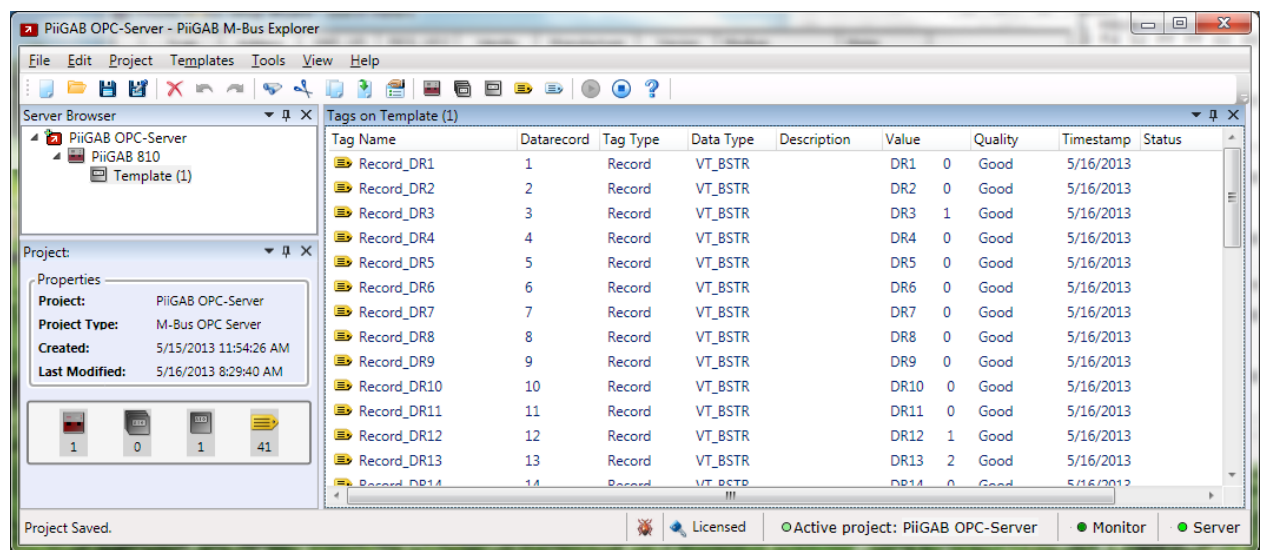
Note:

If the *Server* and monitor led is steady green and not red then one or more OPC-clients are using the OPC-server. You need to terminate these OPC-clients before you start the monitor.

4. Go to Tools menu and click on *Start monitor* or press the  button



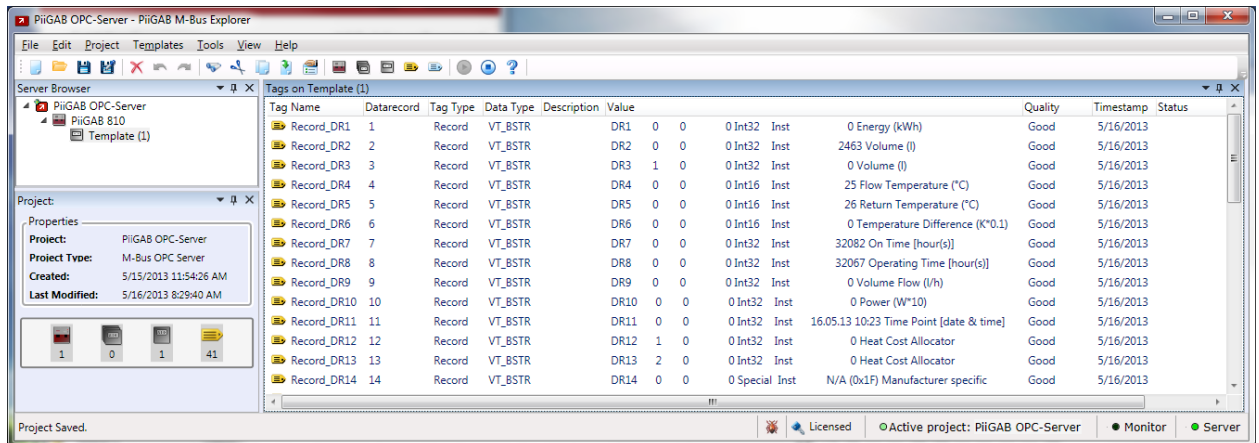
5. Wait for *Quality* cell for each OPC-tag to change to *Good*



**Note:**

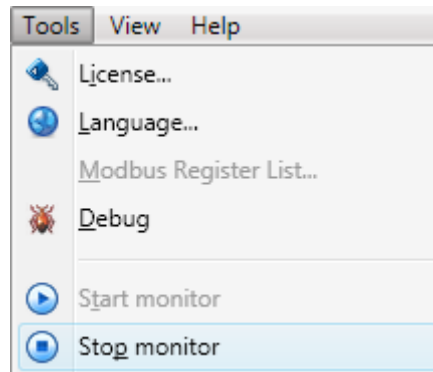
If you don't have quality *Good* on your OPC-tags then you have some configuration errors. Check the primary address, IP-address, port number and protocol. You can also test the communication with the M-Bus master and M-Bus meter with *PiiGAB M-Bus Setup Wizard*.

## 6. Expand the *Value* column so you can see the entire value of each OPC-tag



You can now see what is possible to obtain from each OPC-tag in the M-Bus meter.

## 7. Go to Tools menu and click on *Stop monitor* or press the button



## 6. Select those OPC-tags you want to use for your site

The Browse template recently used has only one purpose: **To show you what is possible to read from the M-Bus meter.** It's not recommended to use it in the OPC-server on your site. You should use the information obtained in the template to configure a new meter in PiiGAB Explorer which suits your site.

Tag Name	Daterecord	Tag Type	Data Type	Description	Value	Quality
Record_DR1	1	Record	VT_BSTR	DR1	0 0 0 Int32 Inst 0 Energy (kWh)	Good
Record_DR2	2	Record	VT_BSTR	DR2	0 0 0 Int32 Inst 2463 Volume (l)	Good
Record_DR3	3	Record	VT_BSTR	DR3	1 0 0 Int32 Inst 0 Volume (l)	Good
Record_DR4	4	Record	VT_BSTR	DR4	0 0 0 Int16 Inst 25 Flow Temperature (°C)	Good
Record_DR5	5	Record	VT_BSTR	DR5	0 0 0 Int16 Inst 26 Return Temperature (°C)	Good

The picture below shows you how to read the information in the *Value* cell:

Tag Name	Daterecord	Tag Type	Data Type	Description	Value	Quality
Record_DR1	1	Record	VT_BSTR	DR1	0 0 0 Int32 Inst 0 Energy (kWh)	Good

Daterecord
Data type
Value
Represent

This example will use the data records in the meter which represent energy, volume and temperature. For instance data record 1, 2, 4 and 5 are maybe the most important objects to read from the meter. This table will summarize the data records:

Data record	Represents	Data type	Value
1	Energy (kWh)	INT32	0
2	Volume (l)	INT32	2463
4	Flow temperature (°C)	INT16	25
5	Return temperature (°C)	INT16	26

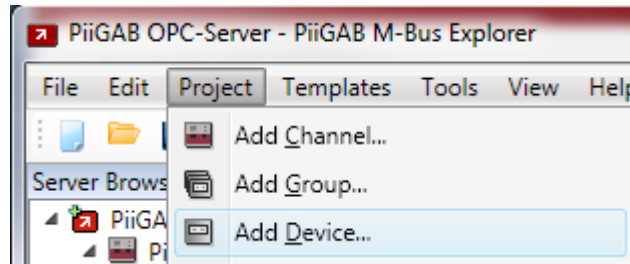
This data is what you need to make a new meter which only contains the data records which are necessary for a typical site.

## 7. Create a configuration that suits your site

From the table in “*Select those OPC-tags you want for your site*” section; a new meter can be used for your site instead of the Browse template meter.

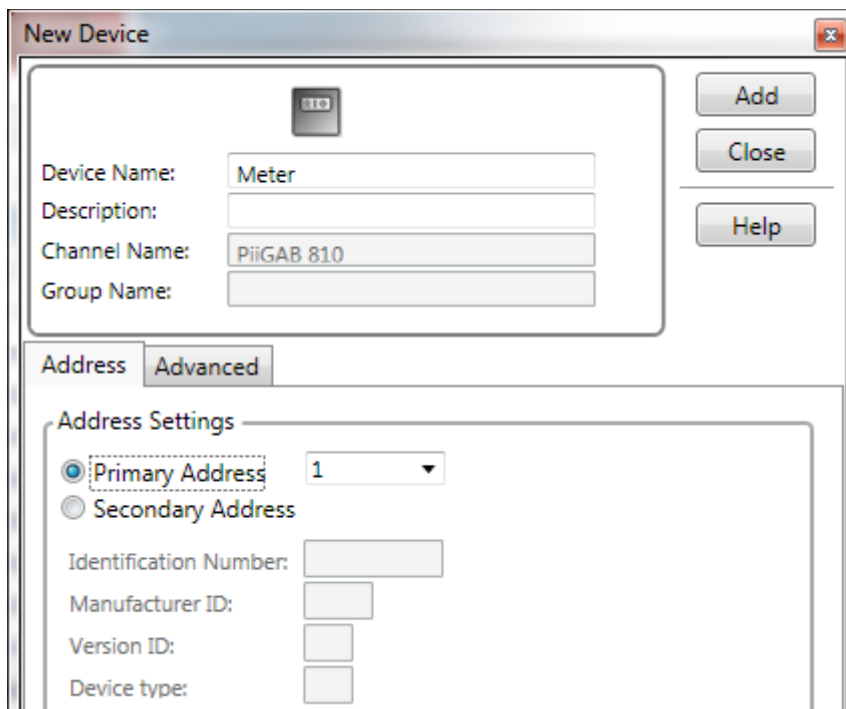
### 7.1 Create a device (meter)

1. Go to the *Project* menu and click on *Add Device...*



You will see a window to configure the device

2. Configure the device as specified in the picture below

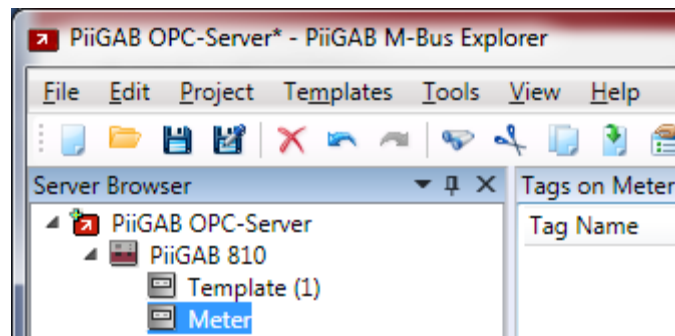


The 'New Device' dialog box contains the following fields and options:

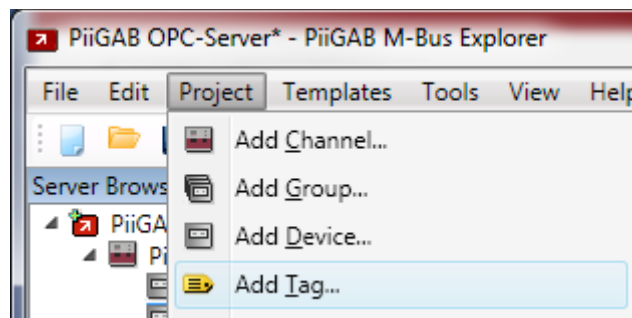
- Device Name:** Meter
- Description:** (empty)
- Channel Name:** PiiGAB 810
- Group Name:** (empty)
- Buttons:** Add, Close, Help
- Address Settings:**
  - Primary Address: 1
  - Secondary Address
  - Identification Number: (empty)
  - Manufacturer ID: (empty)
  - Version ID: (empty)
  - Device type: (empty)

3. Press *Add* to create the device and add it into the project's tree view

4. Select the meter in the tree view

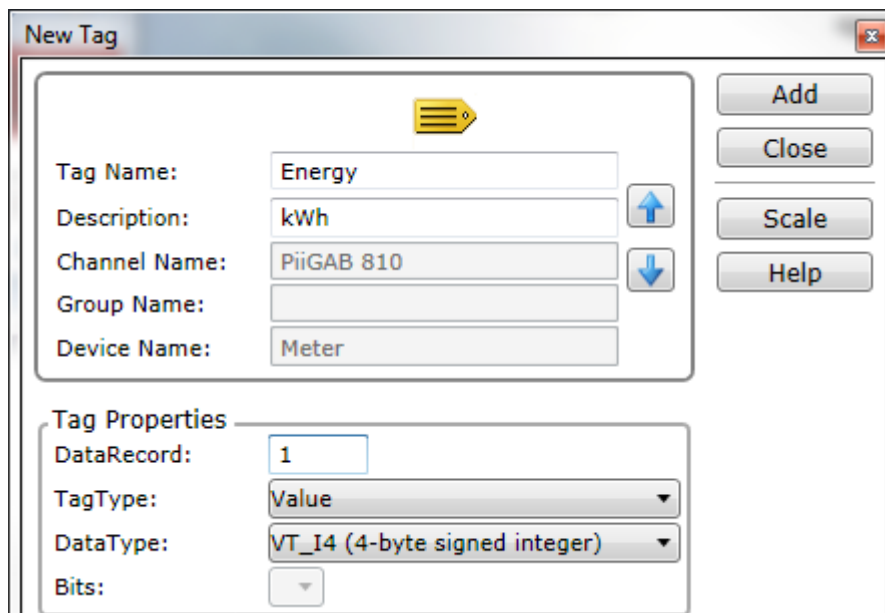


5. Go to *Project* menu and click on *Add tag...*



You will see a window to configure the tag

6. Configure the tag as specified in the picture below



7. Press *Add* to create the tag and add it into the meter's view

8. Repeat the same process from *step 5* for the Volume object

The 'New Tag' dialog box shows the following configuration for a Volume object:

- Tag Name: Volume
- Description: l
- Channel Name: PiiGAB 810
- Group Name: (empty)
- Device Name: Meter
- Tag Properties:
  - DataRecord: 2
  - TagType: Value
  - DataType: VT\_I4 (4-byte signed integer)
  - Bits: (empty)

Buttons on the right: Add, Close, Scale, Help.

9. Also with the temperature objects, notice the data type change

Two side-by-side screenshots of the 'New Tag' dialog box for temperature objects:

- Left screenshot:**
  - Tag Name: Flow temperature
  - Description: C
  - Channel Name: PiiGAB 810
  - Group Name: (empty)
  - Device Name: Meter
  - Tag Properties:
    - DataRecord: 4
    - TagType: Value
    - DataType: VT\_I2 (2-byte signed integer)
    - Bits: (empty)
- Right screenshot:**
  - Tag Name: Return temperature
  - Description: C
  - Channel Name: PiiGAB 810
  - Group Name: (empty)
  - Device Name: Meter
  - Tag Properties:
    - DataRecord: 5
    - TagType: Value
    - DataType: VT\_I2 (2-byte signed integer)
    - Bits: (empty)


Buttons on the right: Add, Close, Scale, Help.

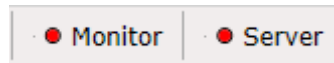
10. The meter list should then be


The screenshot shows the 'PiiGAB OPC-Server - PiiGAB M-Bus Explorer' interface. The 'Tags on Meter' table is displayed as follows:

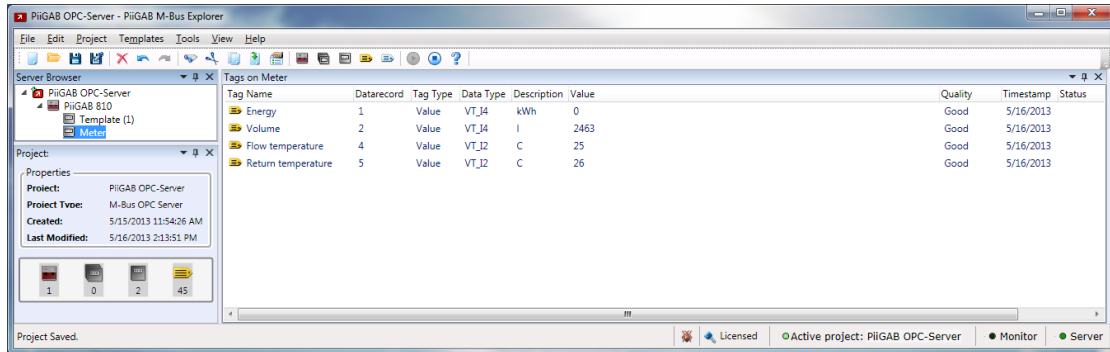
Tag Name	DataRecord	Tag Type	Data Type	Description	Value	Quality	Timestamp	Status
Energy	1	Value	VT_I4	kWh				
Volume	2	Value	VT_I4	l				
Flow temperature	4	Value	VT_I2	C				
Return temperature	5	Value	VT_I2	C				

The interface also shows a 'Server Browser' on the left with a tree view containing 'PiiGAB OPC-Server', 'PiiGAB 810', 'Template (1)', and 'Meter'. The 'Properties' pane shows project details for 'PiiGAB OPC-Server'.

11. Save the project by pressing the  button in the top left corner
12. Make sure the *Server* monitor is steady red in the bottom right corner



13. Start the monitor by pressing the  button
14. Wait for *Quality* cell for each OPC-tag to change to *Good*



You should now have a meter with only those OPC-tags you need for your site. Instead of the entire object's record as value, you only have the actual value.

15. Press the  button to stop the monitor



## 8. Configure number of telegrams for a meter

The data records in an M-Bus meter are located in either one or several telegrams. In a multi-telegram meter the data records will be located in several telegrams. A single-telegram meter has all data records located in the meter's only telegram. The most common data records are usually located in the first telegram.

There is a *Number of telegrams* field for each meter in the PiiGAB Explorer. This can be altered to specify how many telegrams to read for a meter. This field is very important to set. You don't want the client for the PiiGAB Explorer/OPC-server to wait for telegrams you don't need and waste time on.

### 8.1 Theoretic example

Here is an example of an imaginary electric meter with three telegrams.

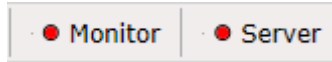
Data record	Represents	Data type	Value
<i>Telegram 1</i>			
1	Power L1 (W)	INT32	1380
2	Power L2 (W)	INT32	1380
3	Power L3 (W)	INT32	1380
4	Voltage L1 (VAC)	INT16	230
5	Voltage L2 (VAC)	INT16	230
6	Voltage L3 (VAC)	INT16	230
<i>Telegram 2</i>			
7	Current L1 (A)	INT16	10
8	Current L2 (A)	INT16	10
9	Current L3 (A)	INT16	10
10	Power factor L1	REAL	0,6
11	Power factor L2	REAL	0,6
12	Power factor L3	REAL	0,6
<i>Telegram 3</i>			
13	Apparent power L1 (VA)	INT32	2300
14	Apparent power L2 (VA)	INT32	2300
15	Apparent power L3 (VA)	INT32	2300
16	Reactive power L1 (Var)	INT32	1840
17	Reactive power L2 (Var)	INT32	1840
18	Reactive power L2 (Var)	INT32	1840


Acquiring power and voltage only requires reading telegram 1. Apparent power requires reading telegram 1, 2 and 3 even if you don't want the data from telegram 1 or telegram 2.






























## 8.2 Find telegrams in meter

The *Browse* template will show how many telegrams there are within its 40 data records. Of course there might be more telegrams and data records beyond the 40 data records in the meter. If there are, the *Browse* template can be altered to display them.

1. Open the *Template (1)* meter
2. Make sure the *Server* monitor is steady red in the bottom right corner



3. Start the monitor by pressing the  button
4. Wait for *Quality* cell for each OPC-tag to change to *Good*

Tags on Template (1)										
Tag Name	Data record	Tag Type	Data Type	Description	Value					Quality
 Record_DR1	1	Record	VT_BSTR	DR1	0 0	0 BCD12	Inst	20674	Energy (Wh*10)	Good
 Record_DR2	2	Record	VT_BSTR	DR2	0 0	0 Variable	Inst	D315	Metrology (firmware) version#	Good
 Record_DR3	3	Record	VT_BSTR	DR3	0 0	0 Int64	Inst	76	Error flags (binary)	Good
 Record_DR4	4	Record	VT_BSTR	DR4	0 0	0 Int8	Inst	59	Power fail counter	Good
 Record_DR5	5	Record	VT_BSTR	DR5	0 0	0 BCD12	Inst	1165412	Total power outage time	Good
 Record_DR6	6	Record	VT_BSTR	DR6	0 0	0 BCD12	Inst	0	Time Point [date & time]  Error: Data error	Good
 Record_DR7	7	Record	VT_BSTR	DR7	0 0	0 Special	Inst	N/A (0x1F)	Manufacturer specific	Good
 Record_DR8	8	Record	VT_BSTR	DR8	0 0	0 Int32	Inst	0	Power (W*0.01)	Good
 Record_DR9	9	Record	VT_BSTR	DR9	0 0	0 Int32	Inst	0	Power (W*0.01) L1	Good
 Record_DR10	10	Record	VT_BSTR	DR10	0 0	0 Int32	Inst	0	Power (W*0.01) L2	Good
 Record_DR11	11	Record	VT_BSTR	DR11	0 0	0 Int32	Inst	0	Power (W*0.01) L3	Good
 Record_DR12	12	Record	VT_BSTR	DR12	0 0	0 BCD4	Inst	2268	Voltage (V*0.1) L1	Good
 Record_DR13	13	Record	VT_BSTR	DR13	0 0	0 BCD4	Inst	2272	Voltage (V*0.1) L2	Good
 Record_DR14	14	Record	VT_BSTR	DR14	0 0	0 BCD4	Inst	2270	Voltage (V*0.1) L3	Good
 Record_DR15	15	Record	VT_BSTR	DR15	0 0	0 BCD4	Inst	0	Current (mA*10) L1	Good
 Record_DR16	16	Record	VT_BSTR	DR16	0 0	0 BCD4	Inst	0	Current (mA*10) L2	Good
 Record_DR17	17	Record	VT_BSTR	DR17	0 0	0 BCD4	Inst	0	Current (mA*10) L3	Good
 Record_DR18	18	Record	VT_BSTR	DR18	0 0	0 BCD4	Inst	5001	Frequency (Hz*0.01)	Good
 Record_DR19	19	Record	VT_BSTR	DR19	0 0	0 Special	Inst	N/A (0x1F)	Manufacturer specific	Good
 Record_DR20	20	Record	VT_BSTR	DR20	0 0	0 Int16	Inst	0	Power factor (*0.001)	Good
 Record_DR21	21	Record	VT_BSTR	DR21	0 0	0 Special	Inst	N/A (0x1F)	Manufacturer specific	Good
 Record_DR22	22	Record	VT_BSTR	DR22	0 0	1 BCD12	Inst	120501000000	Time Point [date & time] End of first	Good
 Record_DR23	23	Record	VT_BSTR	DR23	0 0	1 BCD12	Inst	20674	Energy (Wh*10)	Good
 Record_DR24	24	Record	VT_BSTR	DR24	0 0	0 Special	Inst	N/A (0x1F)	Manufacturer specific	Good
 Record_DR25	25	Record	VT_BSTR	DR25	0 0	2 BCD12	Inst	120420034833	Time Point [date & time] End of first	Good
 Record_DR26	26	Record	VT_BSTR	DR26	0 0	2 BCD12	Inst	20674	Energy (Wh*10)	Good
 Record_DR27	27	Record	VT_BSTR	DR27	0 0	0 Special	Inst	N/A (0x1F)	Manufacturer specific	Good
 Record_DR28	28	Record	VT_BSTR	DR28	0 0	3 BCD12	Inst	680420034518	Time Point [date & time] End of first	Good
 Record_DR29	29	Record	VT_BSTR	DR29	0 0	3 BCD12	Inst	20674	Energy (Wh*10)	Good

In the picture above, notice *data record 7, 19, 21, 24* and *27*. The *0x1F* text string for these data records indicates that there are more data records in the next telegram. It's therefore possible to calculate how many telegrams are necessary to obtain a specific data record. For example data record 23 requires four telegrams.

There is also a way to see when there are no more telegrams in the meter. The picture bellow shows another meter.

Tag Name	Datarecord	Tag Type	Data Type	Description	Value	Quality
Record_DR28	28	Record	VT_BSTR	DR28	1 0 1 Int32 Inst 0 Energy (kWh*10)	Good
Record_DR29	29	Record	VT_BSTR	DR29	1 0 1 Int32 Inst 0 Volume (!*100)	Good
Record_DR30	30	Record	VT_BSTR	DR30	0 0 1 Int16 Inst 31.07.13 Time Point [date]	Good
Record_DR31	31	Record	VT_BSTR	DR31	0 0 0 Special Inst N/A (0x0F) Manufacturer specific	Good
Record_DR32	32	Record	VT_BSTR		N/A (Empty)	Good
Record_DR33	33	Record	VT_BSTR		N/A (Empty)	Good
Record_DR34	34	Record	VT_BSTR		N/A (Empty)	Good

The *0x0F* text string in data record 31 indicates that there are no more data records in next telegram. Therefore no more telegrams are within the meter.

Data record with value *0x0F* or *0x1F* is called *MDH* (Manufactory Data Header). This indicates that there is a manufactory specific data section which starts from this point in the telegram.

Sometimes there is no *MDH* present in a telegram. For such a telegram there are no more telegrams within the meter. The picture below shows a meter with no MDH present.

Tag Name	Datarecord	Tag Type	Data Type	Description	Value	Quality
Record_DR1	1	Record	VT_BSTR	DR1	0 0 0 BCD8 Inst 60820317 Fabrication No	Good
Record_DR2	2	Record	VT_BSTR	DR2	0 0 0 BCD12 Inst 1180 Volume (!*0.1)	Good
Record_DR3	3	Record	VT_BSTR		N/A (Empty)	Good

### 8.2.1 Multi telegram meter

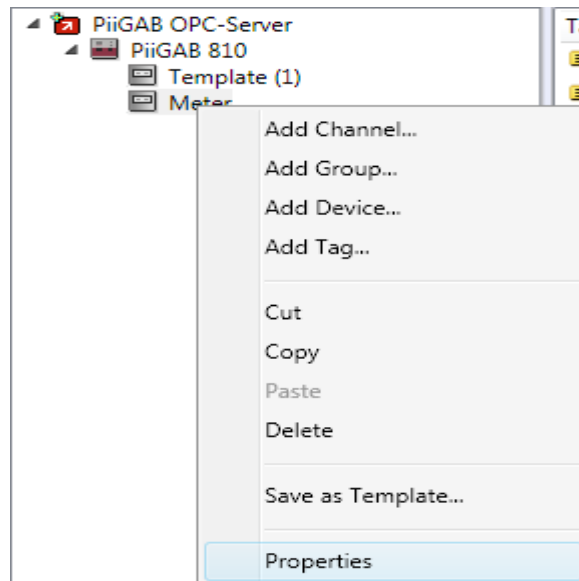
- There are several telegrams.
- Each telegram, except last telegram, have MDH = 0x1F
- Last telegram may or may not have MDH = 0x0F

### 8.2.2 Single telegram meter

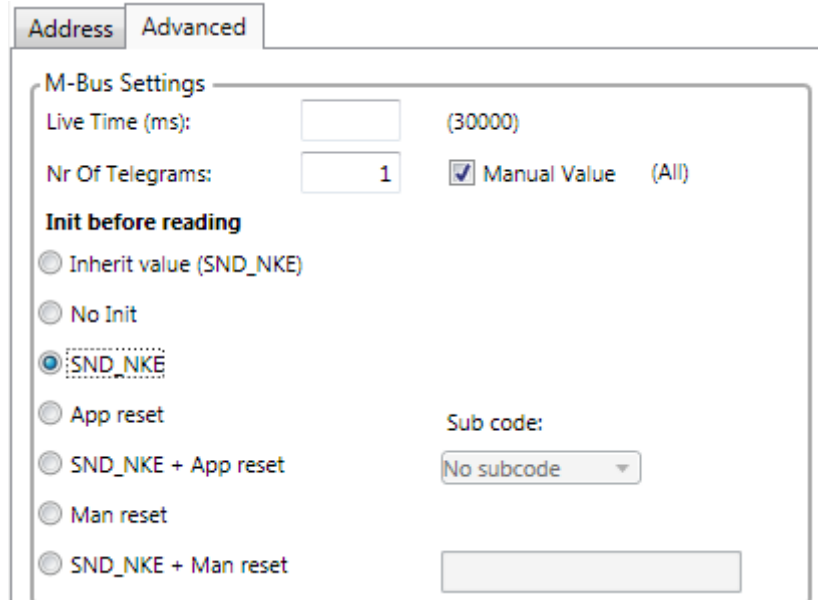
- There is only one telegram.
- The only telegram may or may not have MDH = 0x0F

## 8.3 Set number of telegrams for a meter

1. Right click on the meter in PiiGAB Explorer and select *Properties*



2. Go to the *Advanced* tab
3. Enable *Manual value*
4. Enter the value of telegrams necessary to read, for example 1.
5. Make sure *SND\_NKE* is selected in the *Init before reading* list

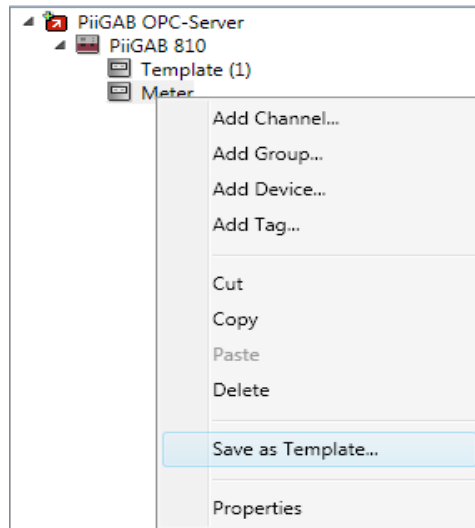


6. Press *OK*

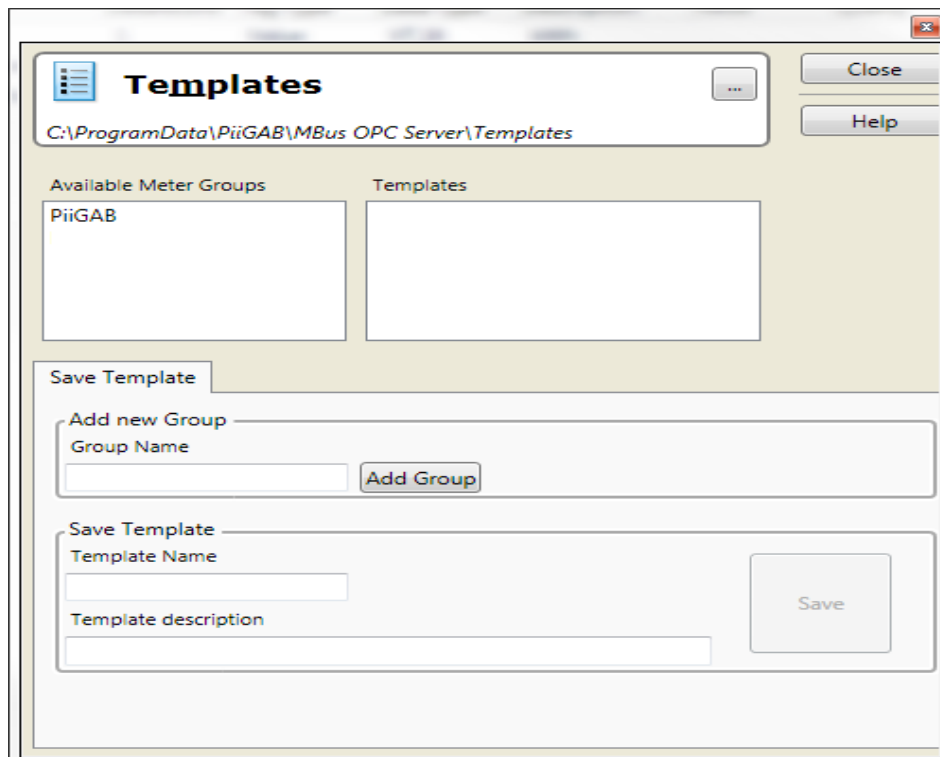
## 9. Save meter as template (optional)

You maybe have several meters of the same type and all read the same data records with the same configuration. In that case you can save the meter as a template and import it as many times as you'll need it.

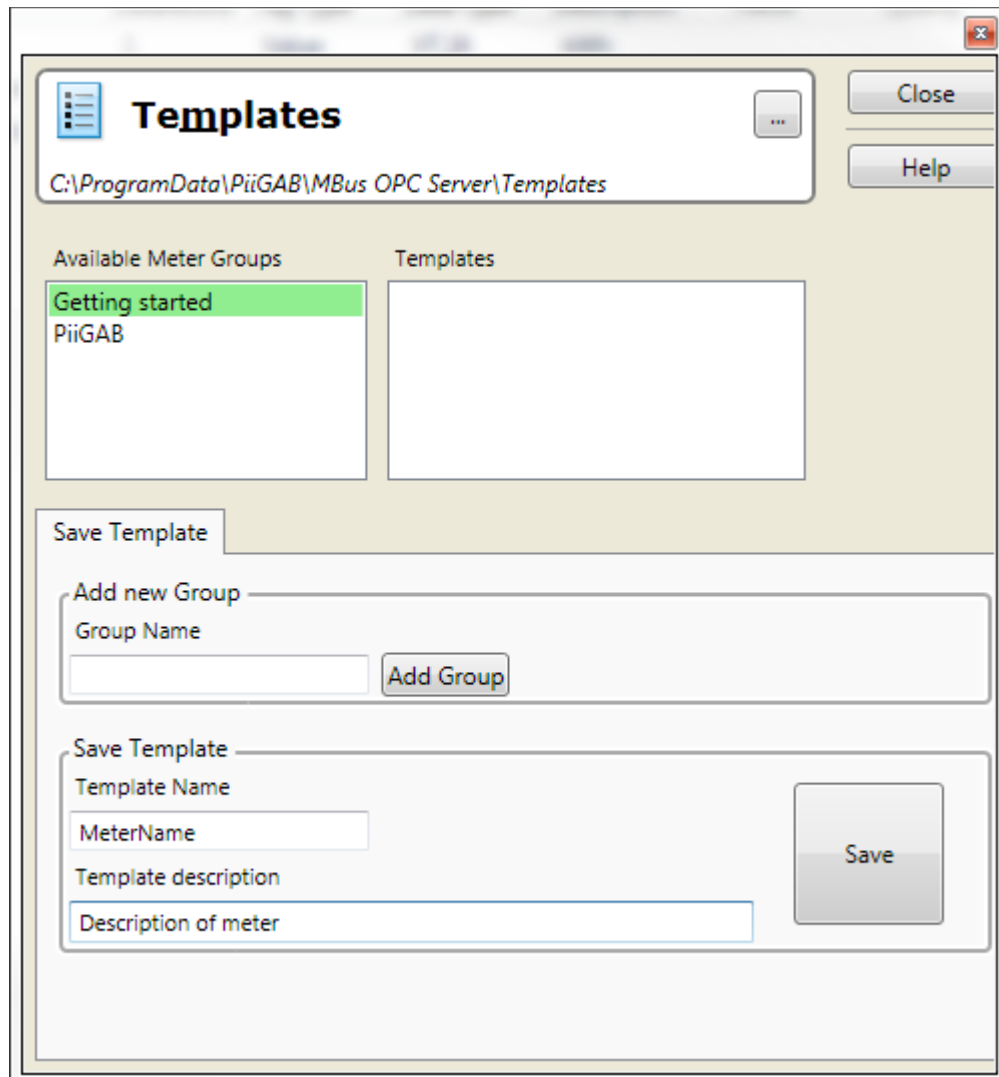
1. Right click on the meter in PiiGAB Explorer and select *Save as Template...*



2. Select the group where you would like to save the template in. For example PiiGAB or create your own group.



3. Give the template a name
4. Optional you can give a description to the template



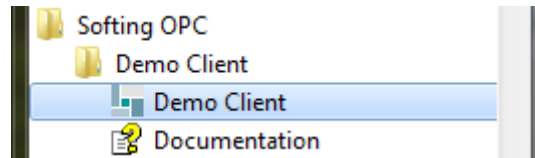
5. Press the Save

You can now import the template for your meter as you did with the Browse template.

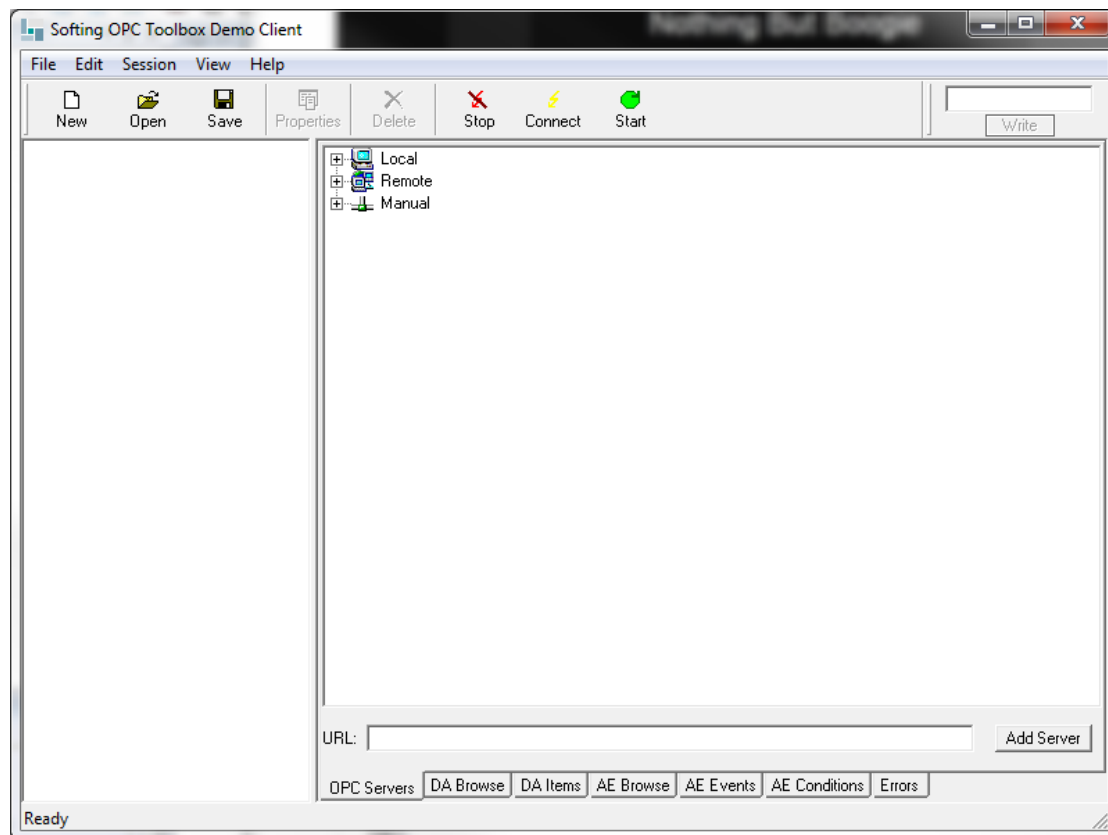
## 10. Test with an OPC-client (optional)

Besides using PiiGAB Explorer's built-in OPC-client you can try reading the M-Bus meter with any OPC-client, for instance *Softing OPC Toolbox Demo Client* from softing.

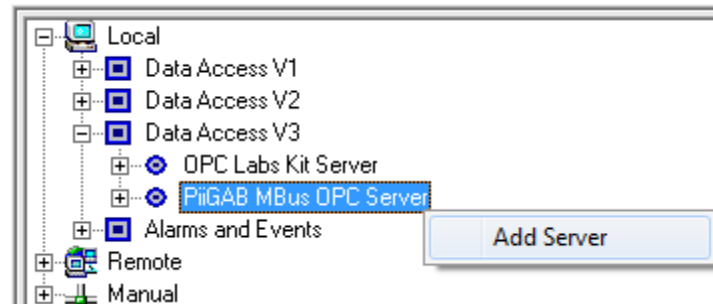
1. From the Window's *Start menu* find the *Softing OPC* folder



2. Run the *Demo Client* application

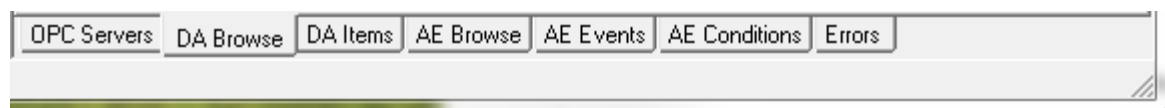


3. Open the *Local* branch and browse to *PiiGAB MBus OPC Server* under the *Data Access V3* node
4. Right click on *PiiGAB MBus OPC Server* and select *Add Server*

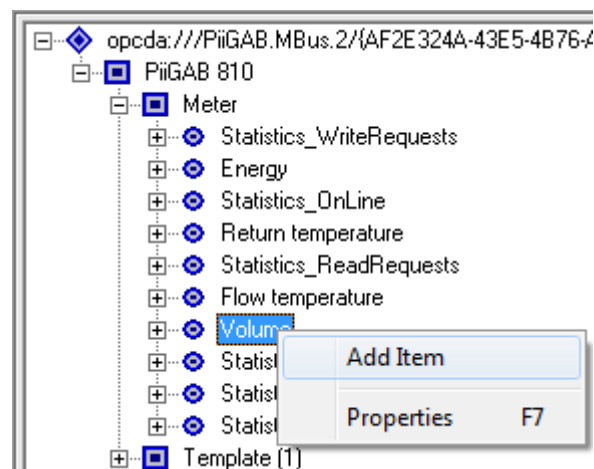


Notice the server was included in the tree structure to the left; a group was attached to it as well.

5. In the bottom of the application click the *DA Browse* tab

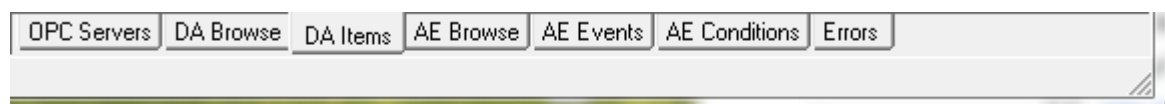


6. Open up the *opcda://PiiGAB.MBus.2/* node and browse to the *Meter* node
7. Right click on the *Volume* node and select *Add Item*



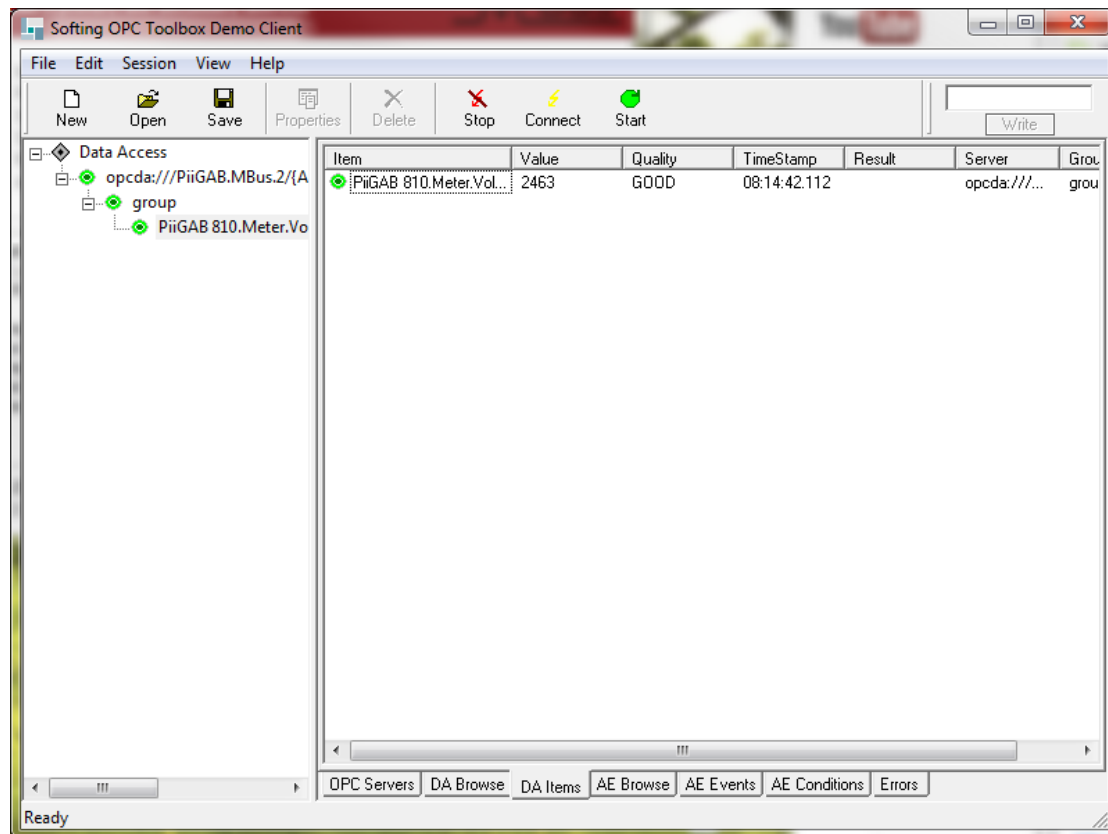
Notice the item was included in the tree structure to the left under the server and the group nodes.

8. In the bottom of the application click the *DA Item* tab





You should then see the value and quality of the *Volume* item



# 11. Appendix

## 11.1 Contacts

### **PiiGAB Processinformation**

Anders Carlssons gata 7  
417 55 Göteborg  
Sweden

Phone + 46 31 55 99 77  
[www.piigab.com](http://www.piigab.com)

### **Distributors**

Please contact our distributors in respective countries:

#### **Germany**

**Relay GmbH**  
Stettiner Str. 38  
33106 Paderborn  
Germany

Phone +49 5251 17670  
[www.relay.de](http://www.relay.de)

#### **Norway**

**Autic Systems AS**  
Stoltenbergs gate 48  
3110 Tønsberg  
Norway

Phone +47 33 30 09 50  
[www.autic.no](http://www.autic.no)

#### **Czech Republic**

**Papouch store s.r.o.**  
Strasnicka 3165/1b  
102 00 PRAGUE 10  
Czech Republic

Phone +420 267 314 267-8  
[www.papouch.com](http://www.papouch.com)