



## IBM Integration Bus Manufacturing Pack

Factory Publication Pattern with MQTT  
Generation of simple application to read data from an  
OPC-UA Server and publish it to an MQTT server

Nov 2014

<b>1. INTRODUCTION</b> .....	<b>3</b>
1.1 OBJECTIVE .....	3
1.2 IIB MANUFACTURING PACK FACTORY PUBLICATION PATTERN .....	3
<b>2. LAB ENVIRONMENT</b> .....	<b>5</b>
<b>3. START THE OPC-UA SERVER</b> .....	<b>6</b>
<b>4. GENERATE THE FACTORY PUBLICATION APPLICATION</b> .....	<b>7</b>
<b>5. DEPLOY AND TEST THE FACTORY PUBLICATION APPLICATION</b> .....	<b>14</b>

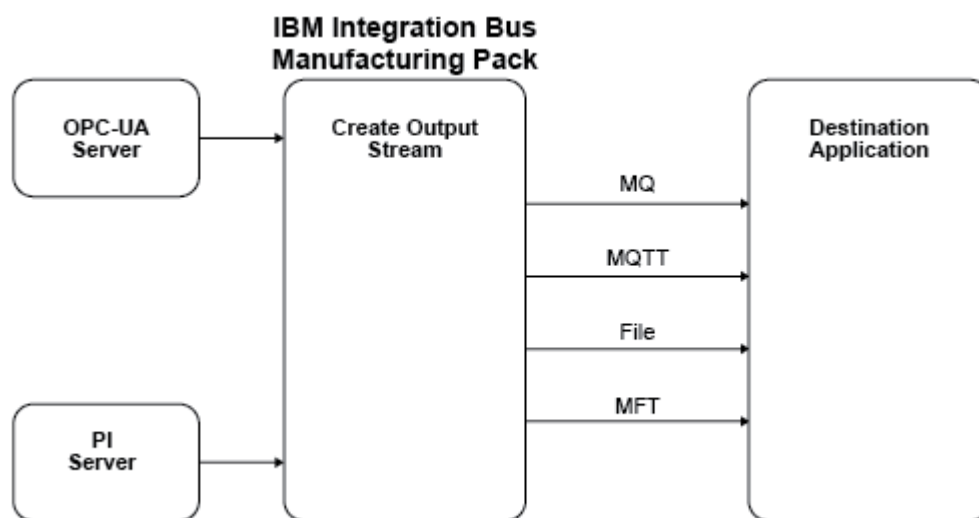
# 1. Introduction

## 1.1 Objective

This lab will show you how to get started quickly with the Manufacturing Pack by using the Factory Publication pattern to create a simple application to read data from an OPC-UA Server and publish it to an MQTT server.

## 1.2 IIB Manufacturing Pack Factory Publication Pattern

The Manufacturing Factory Publication pattern generates an application that takes information from an OPC-UA or PI Server and retransmits this information over a selected transport for use in another application.



Through pattern parameters, the generated application supports the retransmission of OPC-UA and PI data over a selection of transports. In all cases, customization options are provided in the form of a custom map that can be modified after pattern creation to change the format of the output data.

The following output transports are supported:

### MQTT output

Every message that is received by the OPC-UA or PI input nodes in the generated application is published on a chosen topic to the chosen MQTT Server.

### MQ output

Every message that is received by the OPC-UA or PI input nodes in the generated application is placed on a specified WebSphere MQ queue.

### File - Real-time output

Every message that is received by the OPC-UA or PI input nodes in the generated application generates a single file in the specified output directory. These files are time stamped and any attempt to write to a file that already exists results in an exception being thrown in the flow.

**File - Batched output**

Every message that is received by the OPC-UA or PI input nodes in the generated application is treated as a record in the specified output file in the specified output directory. During pattern creation, the user is asked how many records to place in the output file before the file is completed. If the file that is being written already exists, the existing file is archived by using a time stamp.

**MFT - Real-time output**

Every message that is received by the OPC-UA or PI input nodes in the generated application generates a single file in the specified output directory by using MFT. These files are time stamped and any attempt to write to a file that already exists results in that file being overwritten.

**MFT - Batched output**

Every message that is received by the OPC-UA or PI input nodes in the generated application is treated as a record in the specified output file in the specified output directory by using MFT. During pattern creation, the user is asked how many records to place in the output file before the file is completed. The user is also asked whether the file, if it already exists, should be overwritten.

If this option is selected during pattern creation, any of the listed solutions support the logging of every emitted message to an MQ queue.

## 2. Lab Environment

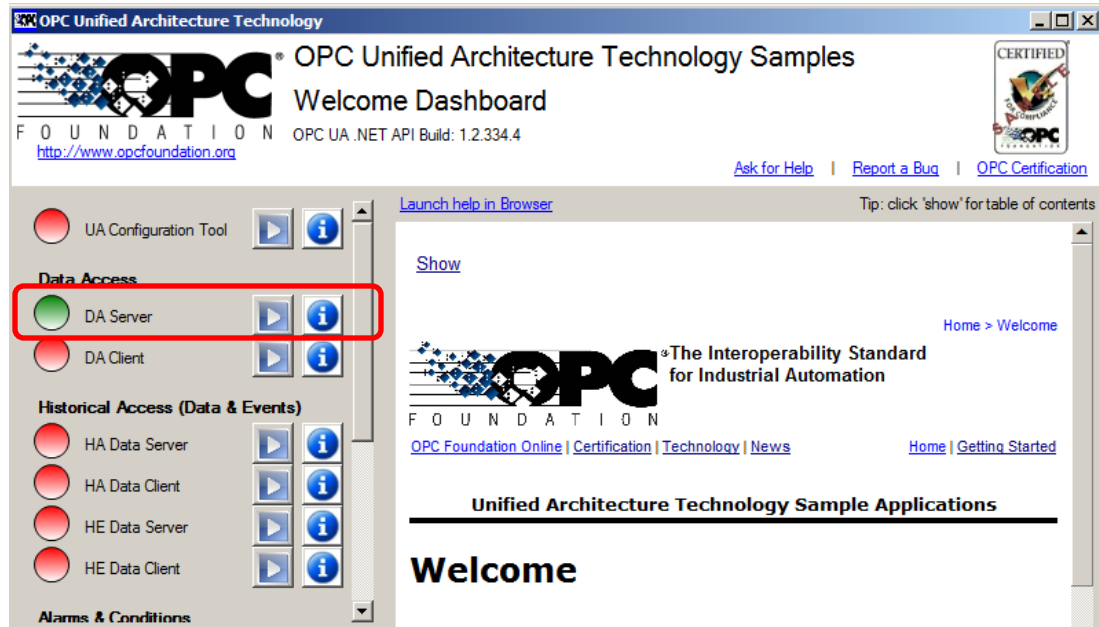
The lab environment consists of an IBM Integration Bus v9.0.0.1 installation with the Manufacturing Pack already installed.

In addition, you will need an OPC-UA Server. In this document, we are using the one included with the Sample Applications from the OPC Foundation (<https://opcfoundation.org/developer-tools/developer-kits-unified-architecture/sample-applications/>). If you use another one, you will need to change the configuration accordingly.

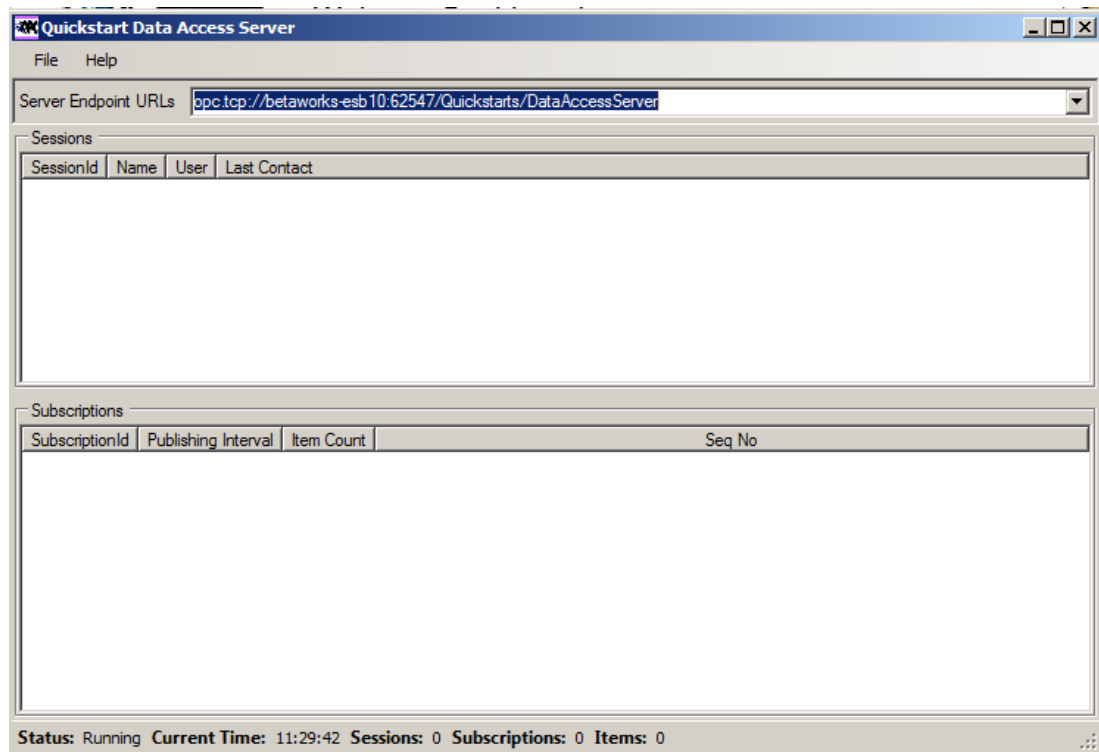
You will also need access to an MQTT server. The one used here is a demonstration one at [messagesight.demos.ibm.com](http://messagesight.demos.ibm.com).

### 3. Start the OPC-UA Server

1. Start the DA Server...

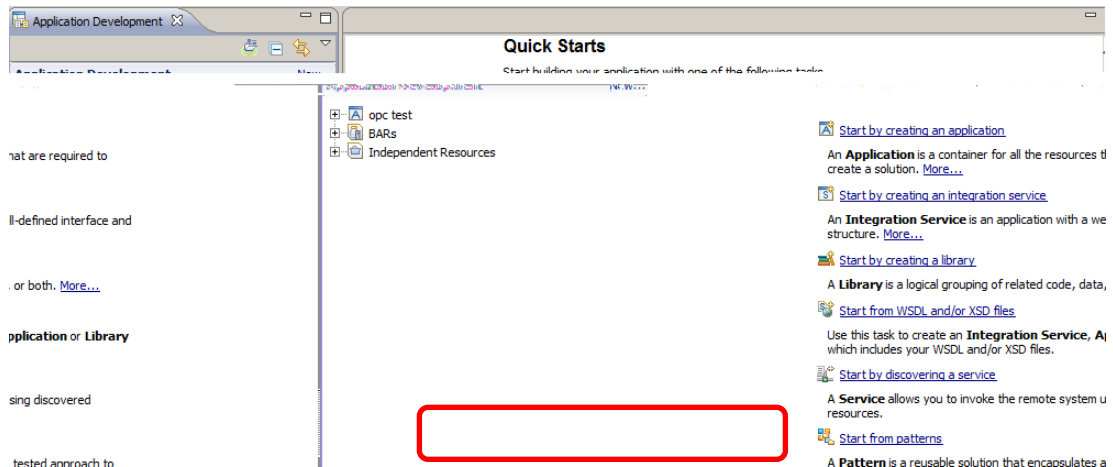


2. ... and make a note of the endpoint URL of the Quickstart Data Access Server.

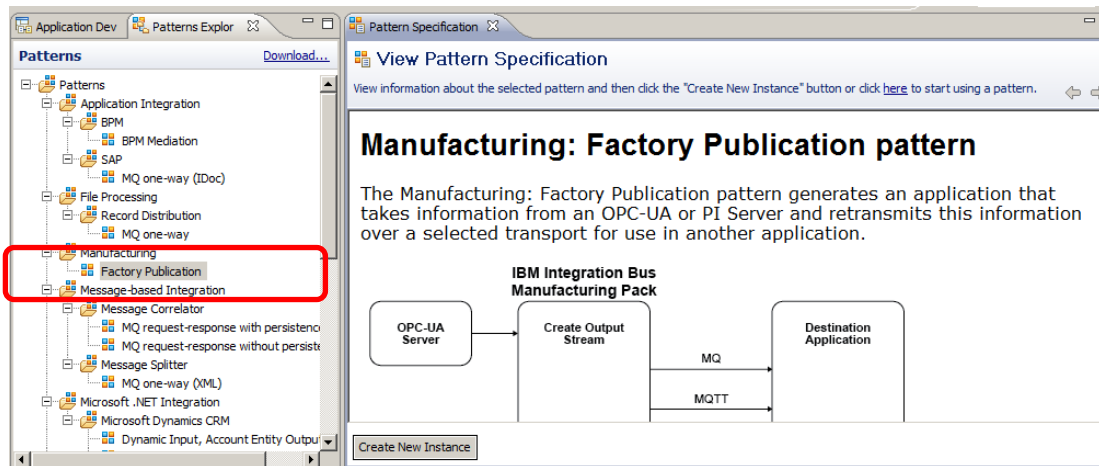


## 4. Generate the Factory Publication Application

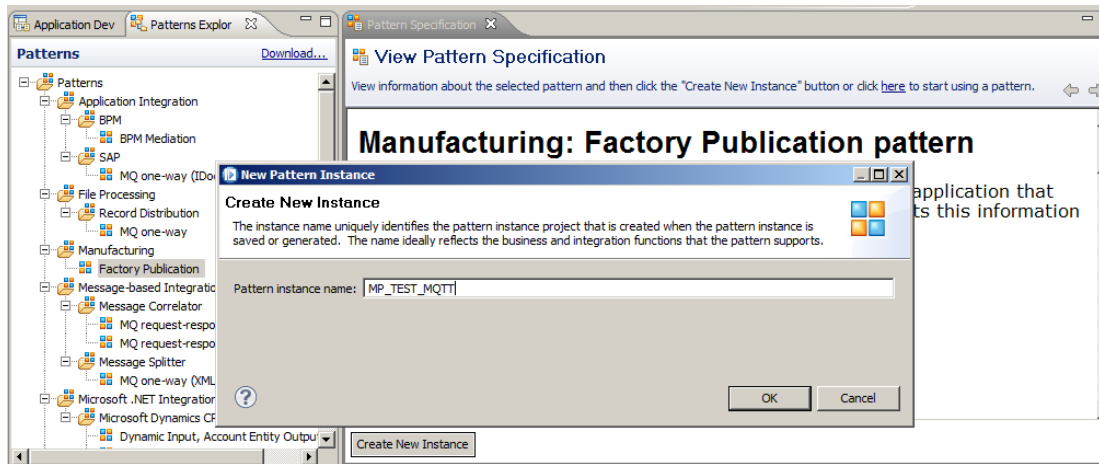
1. Start the IBM Integration Toolkit.
2. From the *Quick Starts* menu click on the *Start from patterns* option.



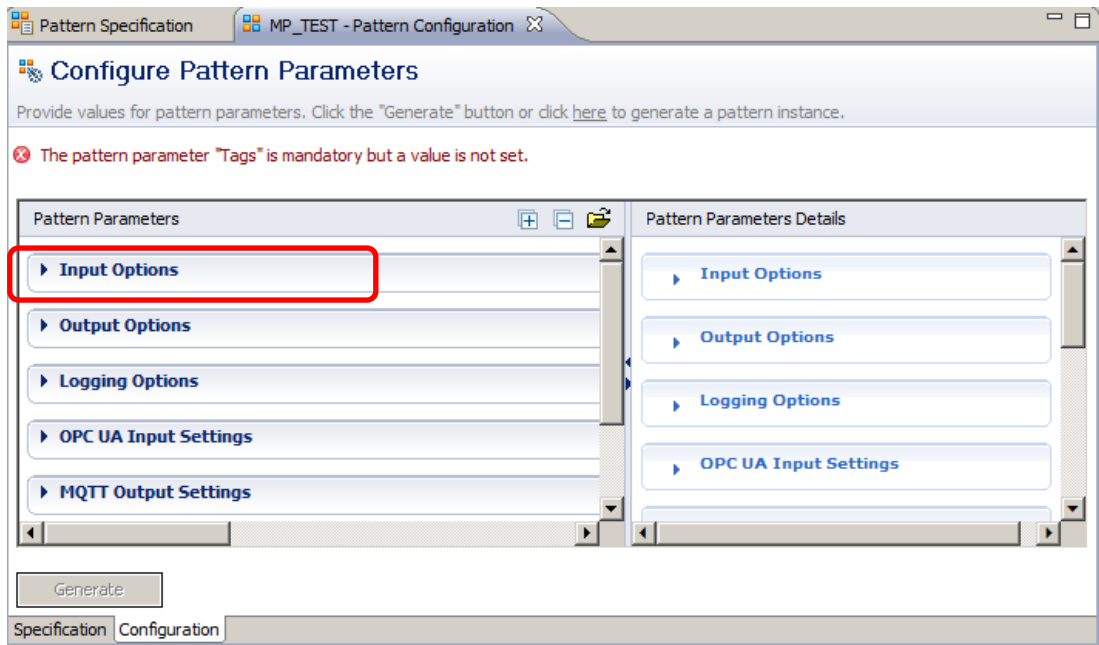
3. From Patterns Explorer, click *Factory Publication* under *Manufacturing*, then click on the *Create New Instance* button in the right-hand pane.



- Provide a name for this new instance and then click *OK*.



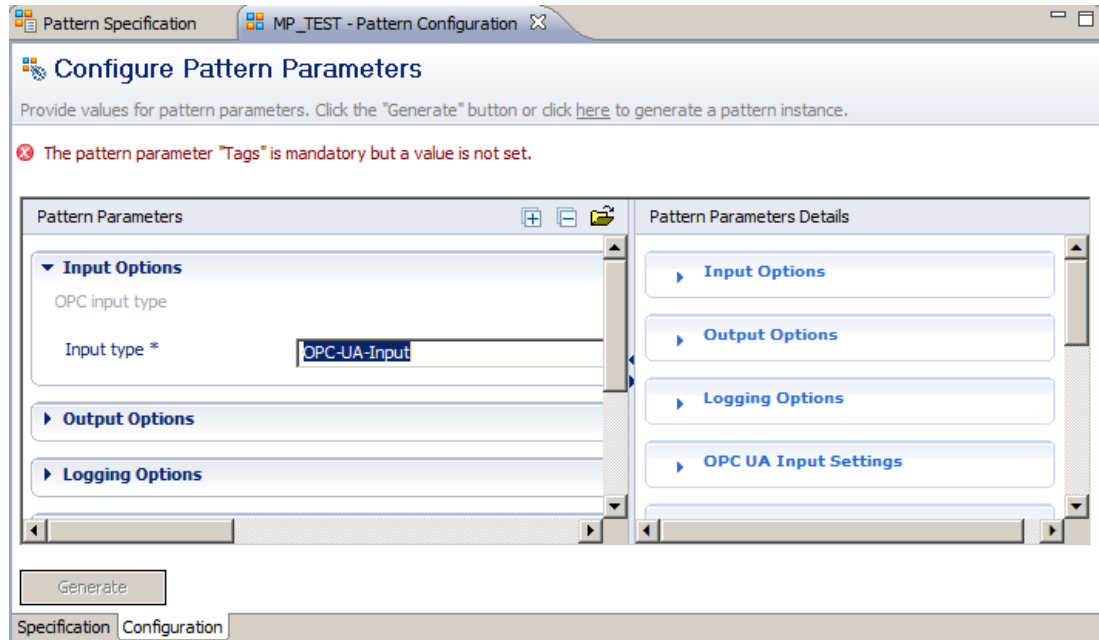
- Under *Configure Pattern Parameters*, expand *Input Options*.



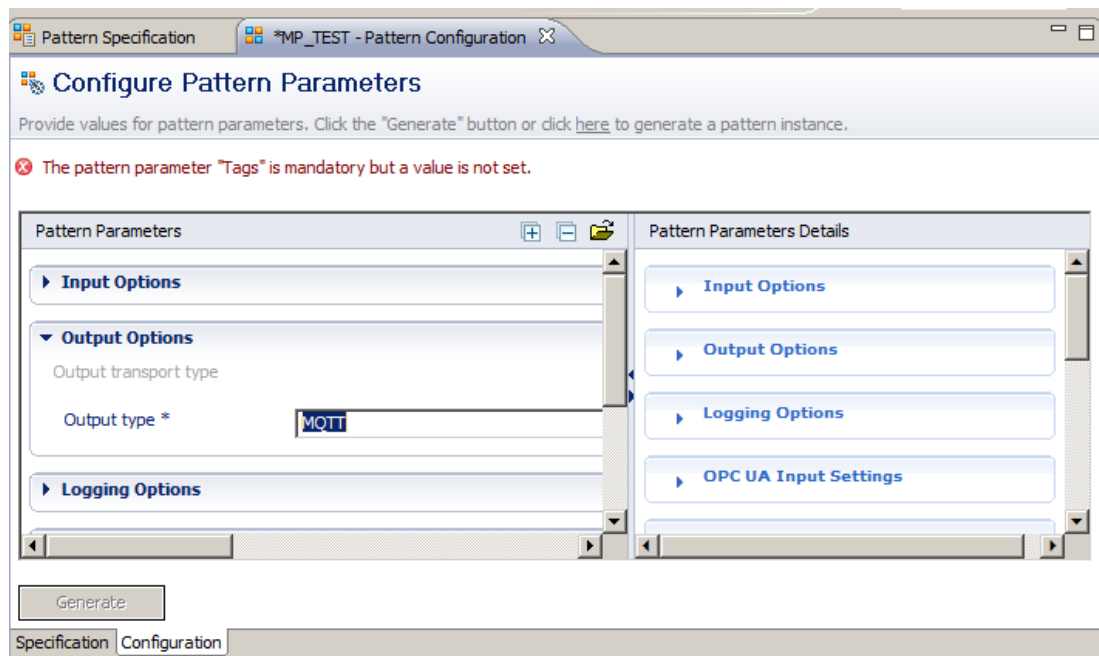


6. There are two input types:
- *OPC-UA-Input*
  - *PI-Input*

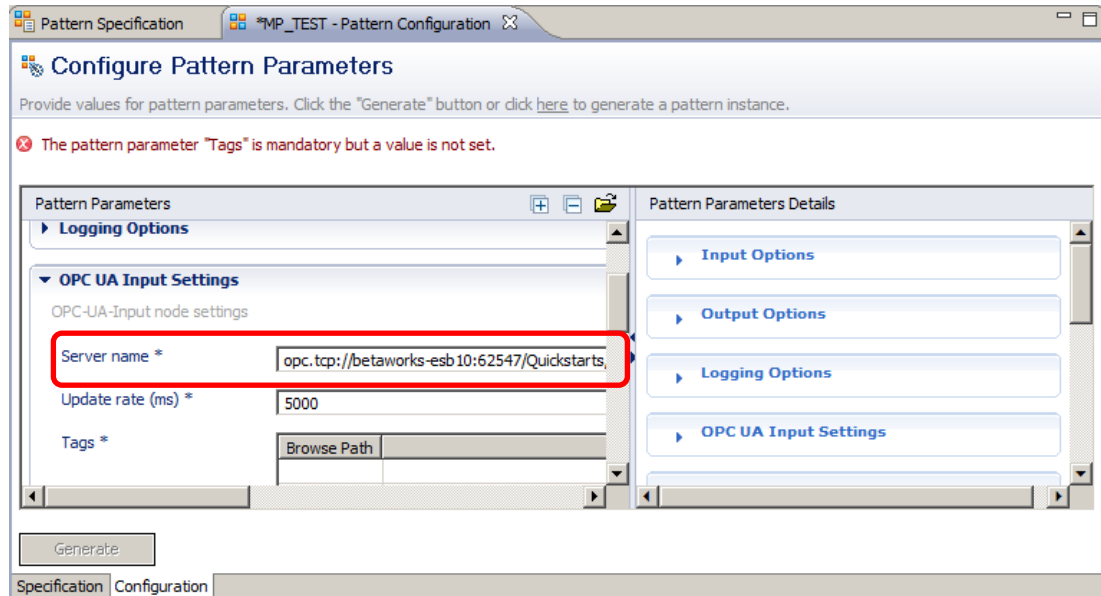
Select the first option.



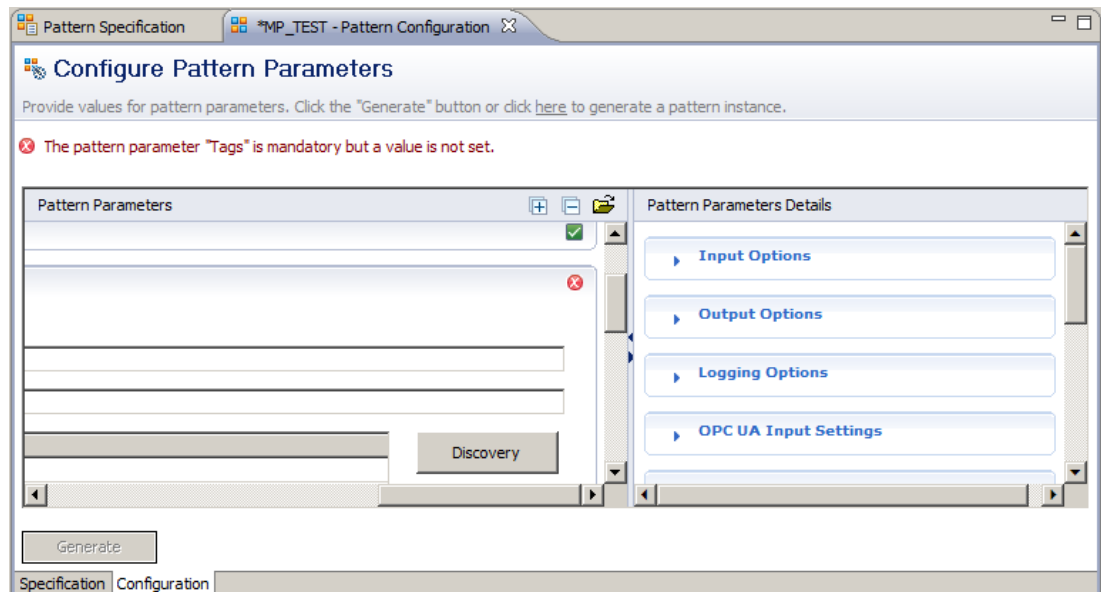
7. Expand Output Options and choose the *MQTT* output type.



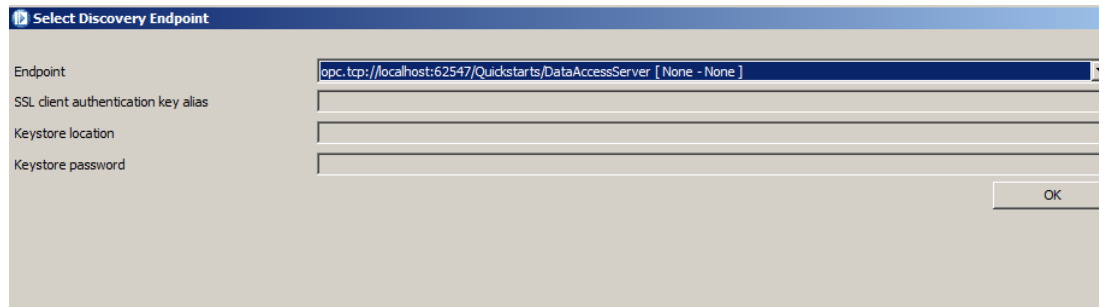
8. Now expand *OPC UA Input Settings* and enter the URL of the Quickstart Data Access Server, noted in section 3, in the *Server name* field.



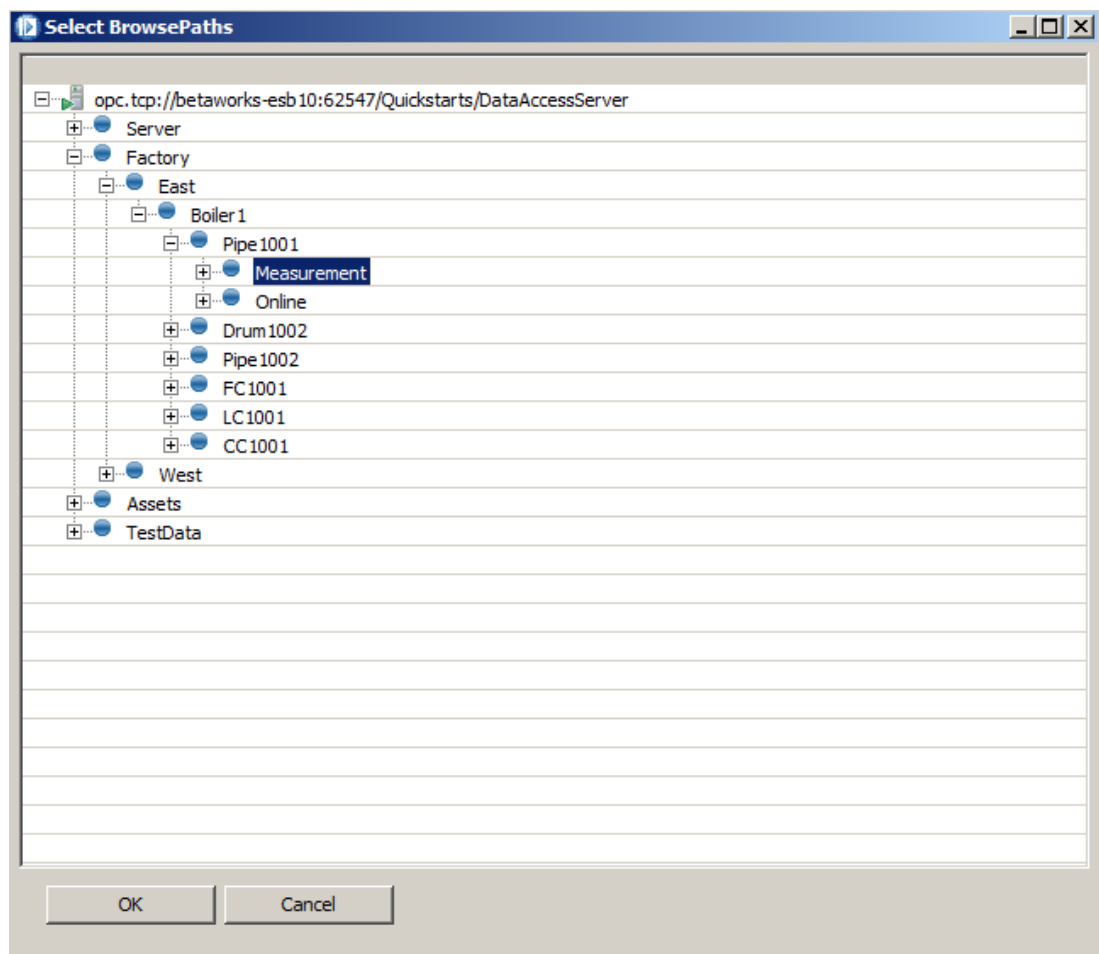
9. Scroll the pane to the right and click the *Discovery* button.



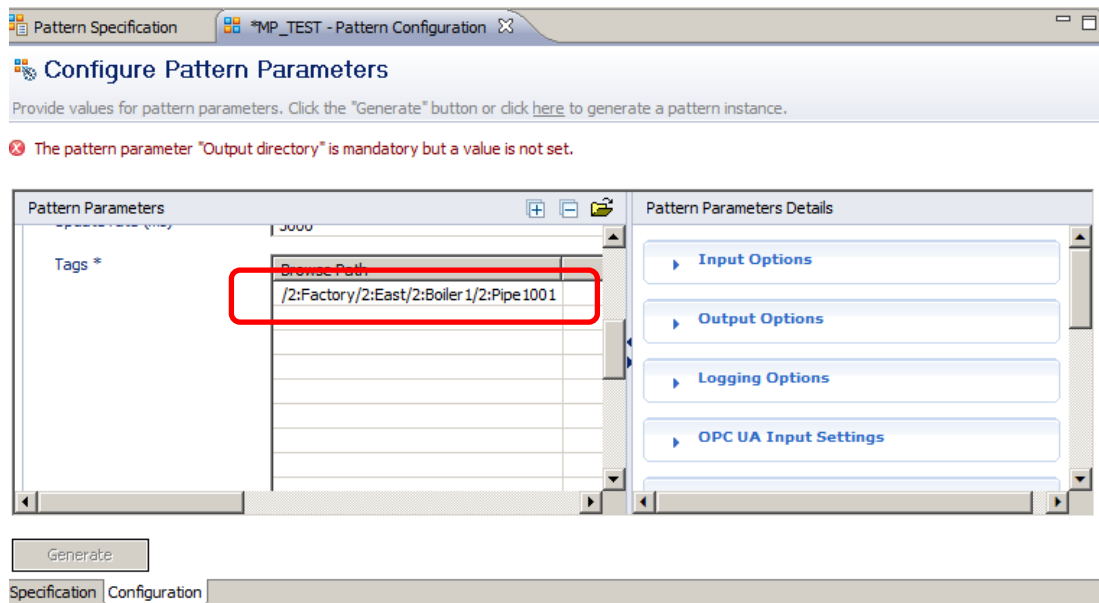
- Click **OK** on the resulting pop-up to confirm the endpoint. There is an option to configure SSL authentication, but this is not being used for this exercise.



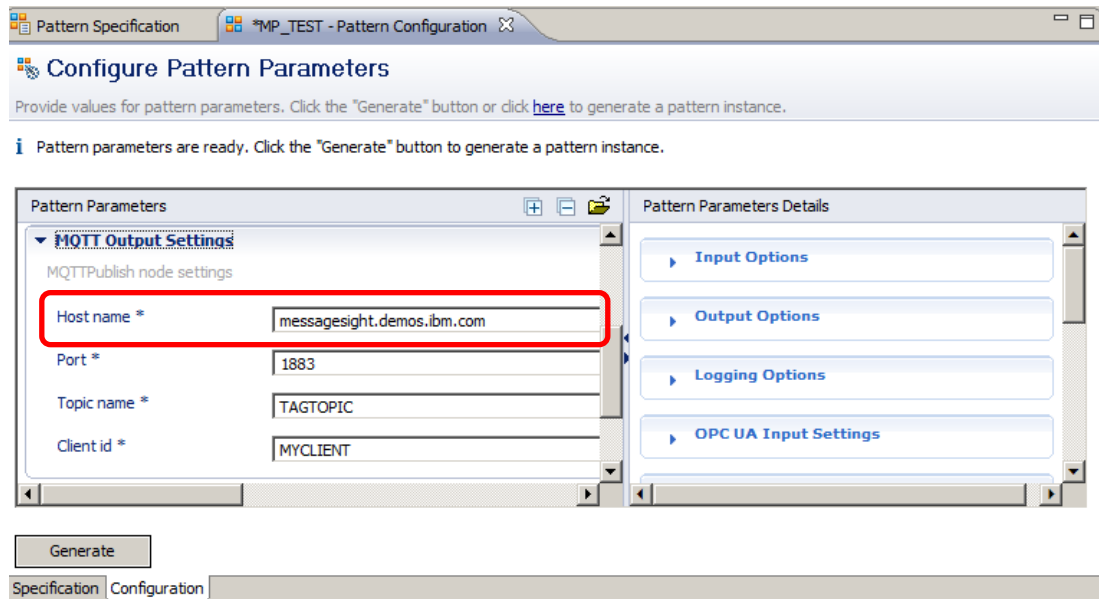
- You should now be presented with paths to data on the OPC-UA DA server. Expand the options and select the one shown below.



12. This action should add the *Browse Path* to the *OPC UA Input Settings*.

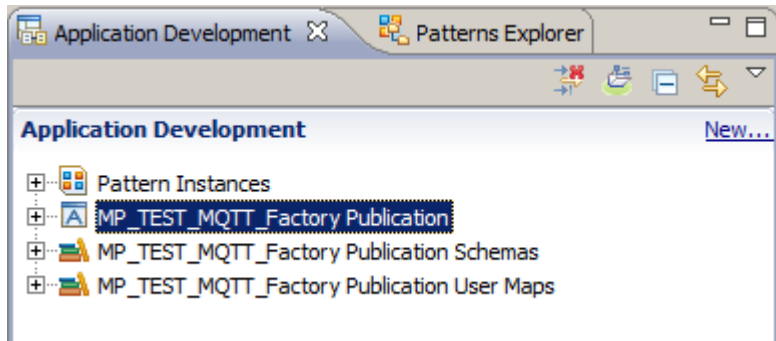


13. Expand the *MQTT Output Settings* and enter the address of the MQTT server to which the data is to be published, in *Host name*. Note the default *Topic name* the data will be published with. Make a note of this for later on.

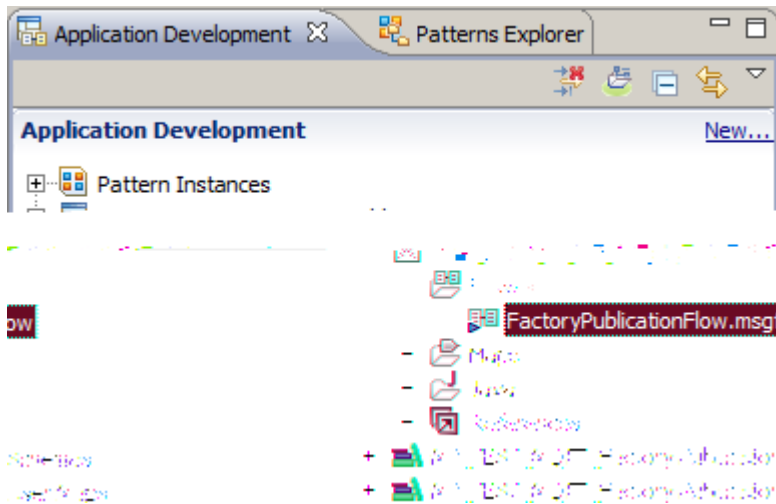


14. Click the *Generate* button.

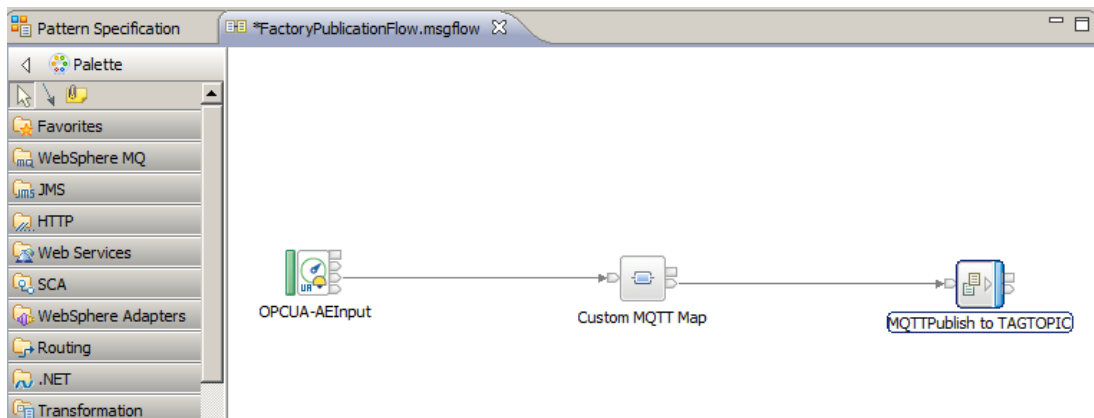
15. You should now see the generated application to the left in the *Application Development* pane.



16. Expand the application and double-click on the message flow.



17. The flow should be like this. Examine the nodes and see how they relate to the configuration settings you provided to the pattern.



## 5. Deploy and Test the Factory Publication Application

1. Before MQTT can be used, the MQTT connector must first be installed on the integration node.

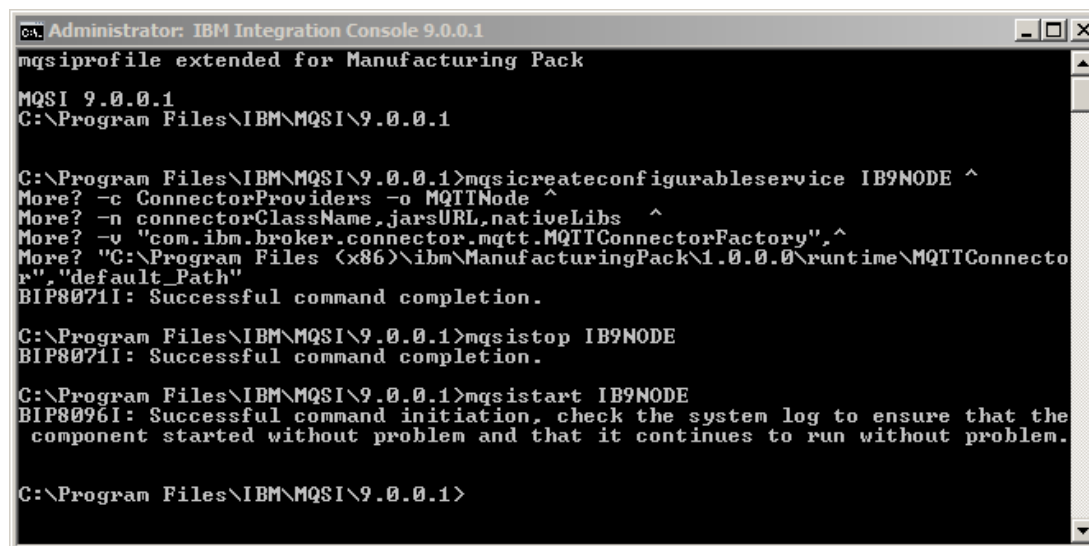
Run the following from the *IIB Integration Console*:

```
mqsicreateconfigurable-service <integration node> ^  
-c ConnectorProviders -o MQTTNode ^  
-n connectorClassName,jarsURL,nativeLibs ^  
-v "com.ibm.broker.connector.mqtt.MQTTConnectorFactory", ^  
"<location of MQTT connector>","default_Path"
```

where

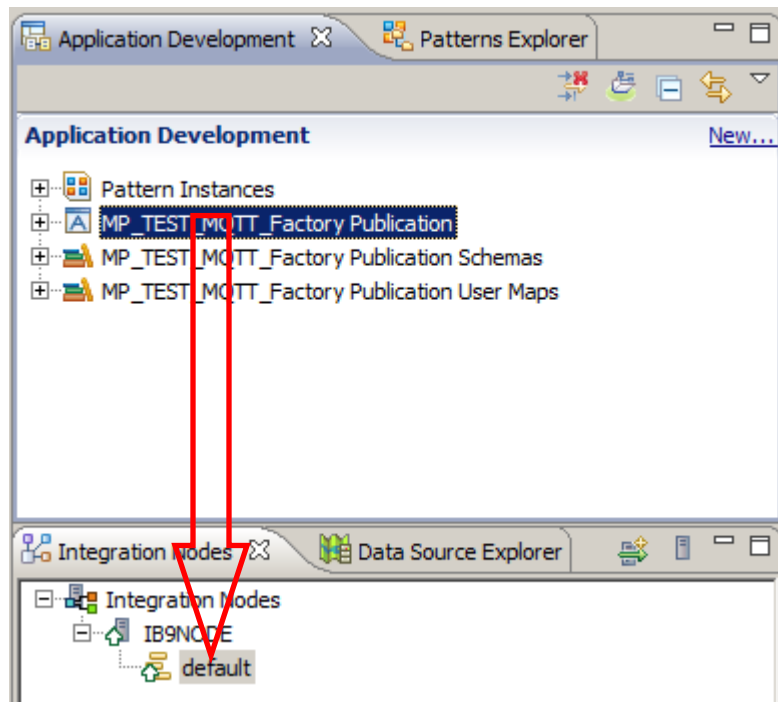
- <integration node> is: *IB9NODE*
- <location of MQTT connector> is: *C:\Program Files (x86)\Ibm\ManufacturingPack\1.0.0.0\runtime\MQTTConnector*

And then stop and re-start the integration node.

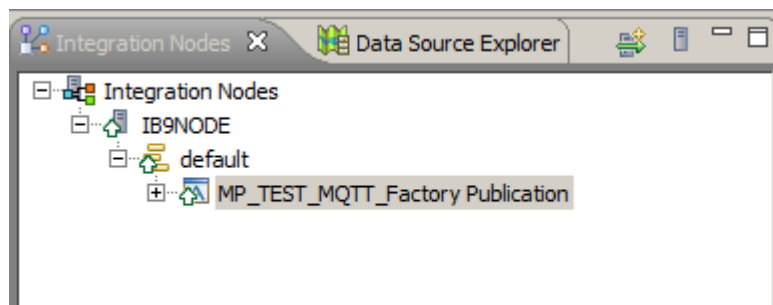


```
Administrator: IBM Integration Console 9.0.0.1  
mqsiprofile extended for Manufacturing Pack  
MQSI 9.0.0.1  
C:\Program Files\IBM\MQSI\9.0.0.1  
  
C:\Program Files\IBM\MQSI\9.0.0.1>mqsicreateconfigurable-service IB9NODE ^  
More? -c ConnectorProviders -o MQTTNode ^  
More? -n connectorClassName,jarsURL,nativeLibs ^  
More? -v "com.ibm.broker.connector.mqtt.MQTTConnectorFactory", ^  
More? "C:\Program Files (x86)\Ibm\ManufacturingPack\1.0.0.0\runtime\MQTTConnector", "default_Path"  
BIP8071I: Successful command completion.  
  
C:\Program Files\IBM\MQSI\9.0.0.1>mqsistop IB9NODE  
BIP8071I: Successful command completion.  
  
C:\Program Files\IBM\MQSI\9.0.0.1>mqsistart IB9NODE  
BIP8096I: Successful command initiation, check the system log to ensure that the component started without problem and that it continues to run without problem.  
  
C:\Program Files\IBM\MQSI\9.0.0.1>
```

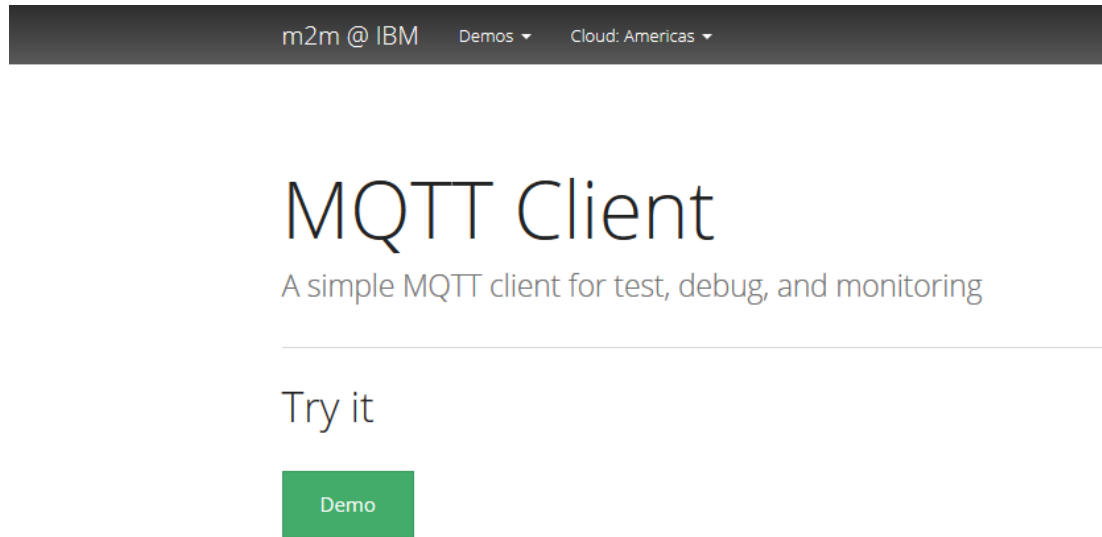
2. Deploy the application by dragging and dropping it onto the integration server as shown:



3. Expand the integration server and you should see the application has been deployed there:



4. Start the MQTT client. In this case, we are using a demonstration one at <http://m2m.demos.ibm.com/utilities.html>



5. Click *Demo* and you will be presented with the following. The default server name and port are the same as the ones that were specified to the Factory Publication pattern earlier, so do not need to be changed.

The screenshot shows the 'Connect' form with the following fields and controls:

- Server:** messagesight.demos.ibm.com
- Port:** 1883
- Connect:** (green button)
- Disconnect:** (red button)
- Client ID:** Client84669
- Username:** (optional)
- Password:** (optional)
- Clean Session:** OFF (grey), ON (green)
- SSL:** OFF (green), ON (grey)



6. Click *Connect* and you should see the screen updated accordingly with connection details.

The screenshot displays a web interface with several sections. At the top, a green banner contains the text "Connected!". Below this is a "Connect" section, followed by a "Subscribe" section. The "Subscribe" section includes a "Topic" input field with the value "planets/earth", a "QOS" dropdown menu set to "0", and a green "Subscribe" button. Below the "Subscribe" section is a "Publish" section. At the bottom, there is a "Log" section with a "Log" button (indicated by a blue circle with the number 1), a "Clear" button, and a "Follow" checkbox. A red box highlights a log entry: "(13:30:18.293) Connected to messagesight.demos.ibm.com:1883".

7. The MQTT topic that data is being published with (see Factory Publication pattern configuration step) is *TAGTOPIC*, so change *Topic* in the *Subscribe* window to this value and click the *Subscribe* button. The screen will be updated with subscription details as shown.

The screenshot shows the MQTT interface with the following elements:

- Connected!** (Green bar)
- Connect** (Blue header)
- Subscribe** (Blue header)
- Topic: TAGTOPIC
- QOS: 0
- Subscribe button (Green)
- TAGTOPIC tag (Blue, highlighted with a red box)
- Publish** (Blue header)
- Log 2 (Blue header)
- Clear button
- Follow checkbox
- Log entries:
  - (13:30:18.293) Connected to messagesight.demos.ibm.com:1883
  - (13:32:11.864) Subscribed to [TAGTOPIC][qos 0] (highlighted with a red box)

8. Entries relating to the selected OPC data will now begin to appear in the log area:

The screenshot shows the MQTT interface with the following elements:

- Log 6 (Blue header)
- Clear button
- Follow checkbox
- Log entries:
  - (13:30:18.293) Connected to messagesight.demos.ibm.com:1883
  - (13:32:11.864) Subscribed to [TAGTOPIC][qos 0]
  - (13:34:38.399) >> [TAGTOPIC] opc.tcp://betaworks-esb10:62547/Quickstarts/DataAccessServer0/2:Factory/2:East/2:Boiler1/2:Pipe1001/2:Measurement1.03E+22014-11-03T13:34:37.284Good
  - (13:34:43.417) >> [TAGTOPIC] opc.tcp://betaworks-esb10:62547/Quickstarts/DataAccessServer1/2:Factory/2:East/2:Boiler1/2:Pipe1001/2:Measurement9.6E+12014-11-03T13:34:42.299Good
  - (13:34:48.526) >> [TAGTOPIC] opc.tcp://betaworks-esb10:62547/Quickstarts/DataAccessServer2/2:Factory/2:East/2:Boiler1/2:Pipe1001/2:Measurement5.7E+12014-11-03T13:34:47.315Good
  - (13:34:53.452) >> [TAGTOPIC] opc.tcp://betaworks-esb10:62547/Quickstarts/DataAccessServer3/2:Factory/2:East/2:Boiler1/2:Pipe1001/2:Measurement1.41E+22014-11-03T13:34:52.331Good

- g. When you have finished testing, stop the application.

