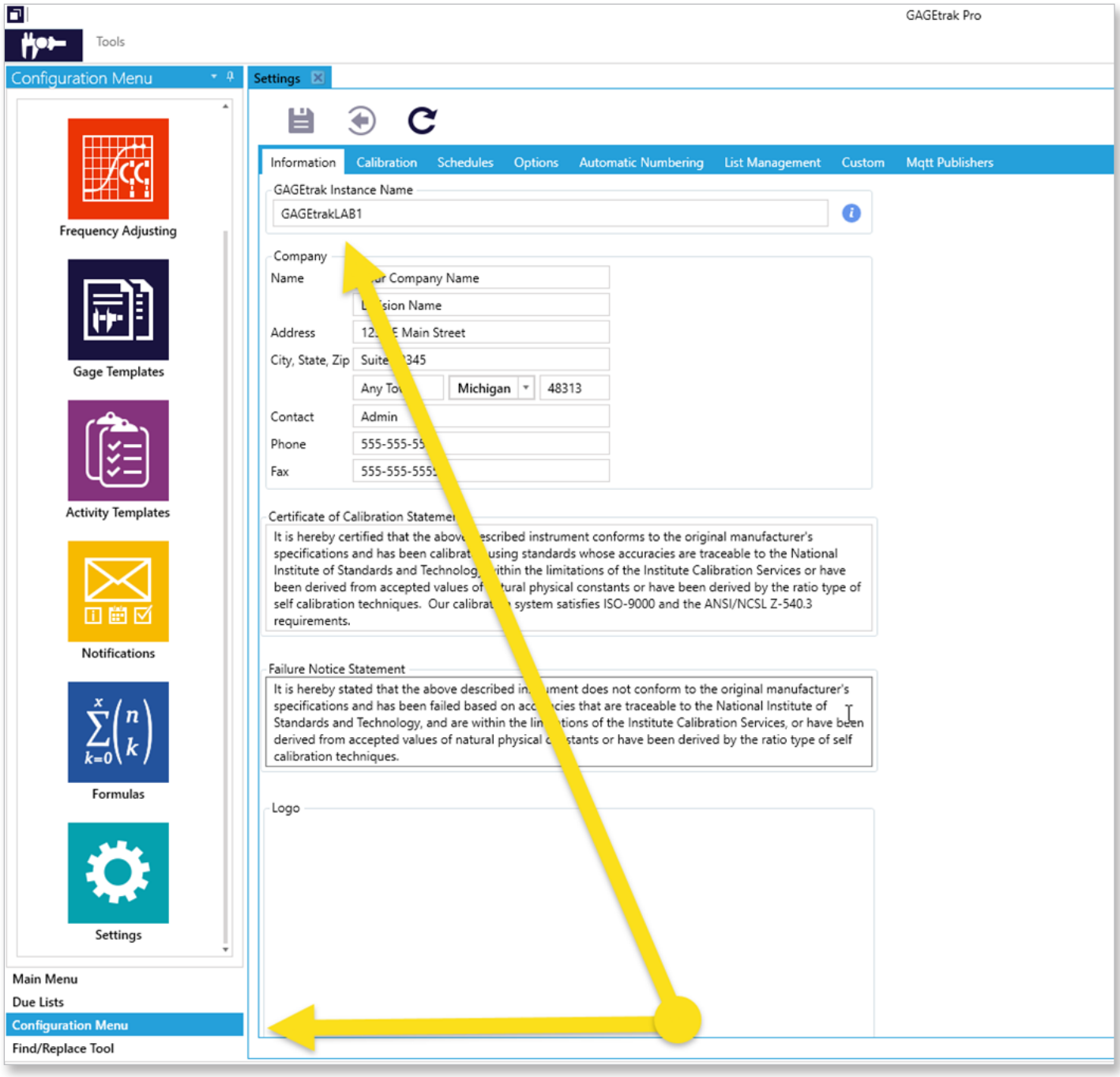
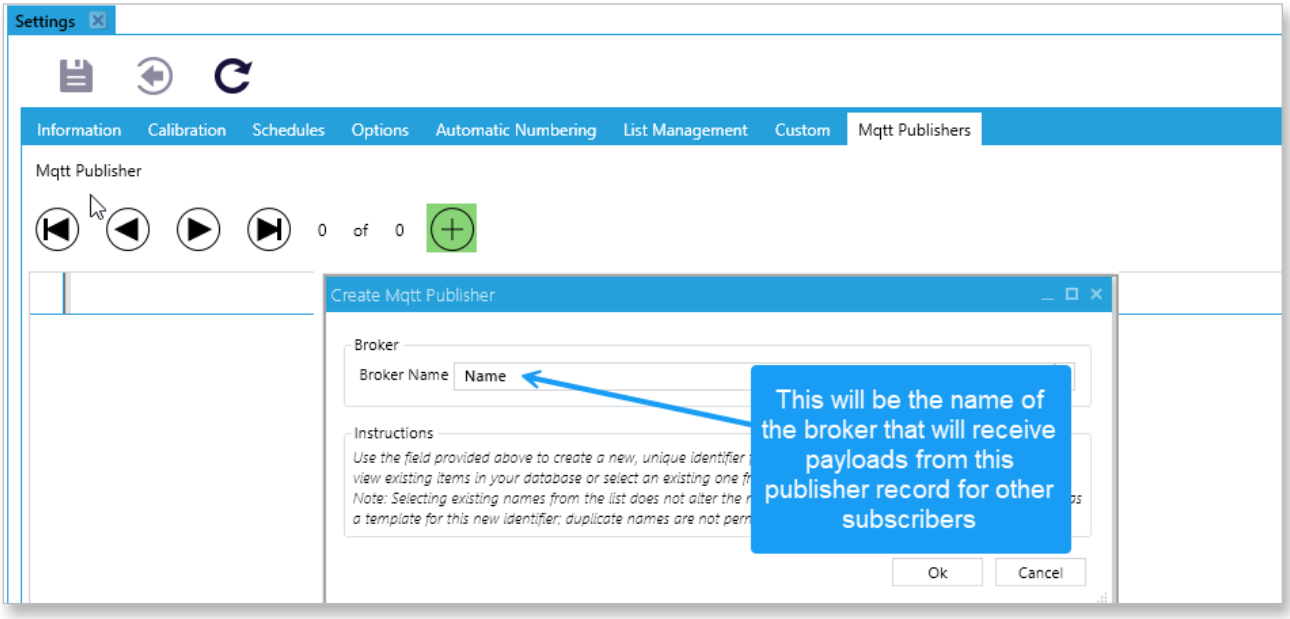
GAGETRAK MQTT PUBLISHER SAMPLE PROJECT GUIDE

Setting up the GAGEtrak Publisher Client

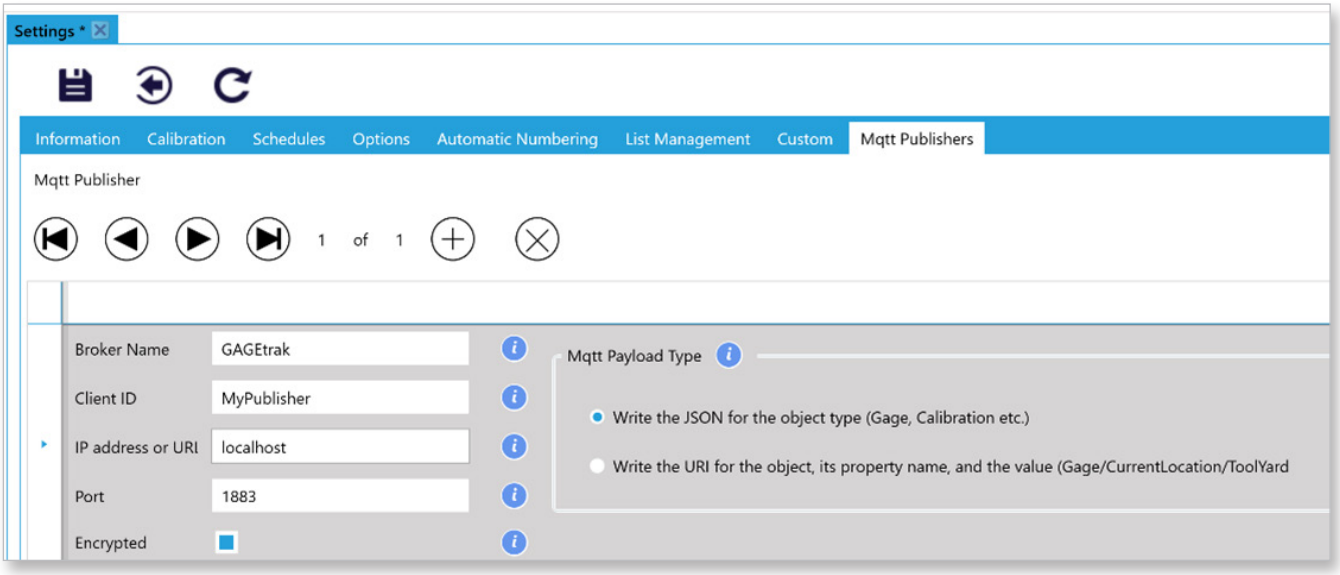
To receive notifications from a GAGEtrak client to your broker, launch the GAGEtrak Pro/Lite application and then select **Settings** from the **Configuration Menu**. In the **Information** tab, enter a **GAGEtrak Instance** name; this is a unique name for your GAGEtrak instance, which will be used for MQTT and by other applications. This will appear at the beginning of the topic in your MQTT output, i.e., GAGEtrakLAB1/Gage/022E-DCAL.



Next, select the **MQTT Publishers** tab. Click the ⊕ icon to add a publisher and enter the name of the broker in the **Create MQTT Publisher** dialog.



Enter a unique **Client ID**, the **IP** or **URI** for the broker, and the **Port** (we will use the same port selected when installing the GAGEtrak MQTT Broker). The **Broker Name** and the **Client ID** can be any arbitrary name, provided they are unique in your environment, i.e., **Broker Name** ***GAGEtrakBroker1*** and **Client ID** ***LAB1GageChanges***. These are simply identifiers that will make it easier to determine the origin of any published data in your MQTT ecosystem. This is especially important if there are multiple brokers and publishers in your environment.

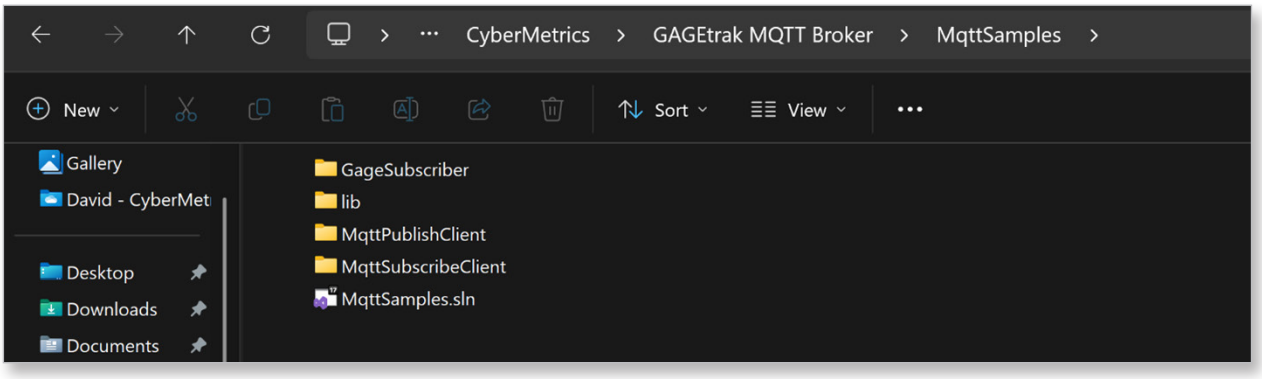


Leave the default **MQTT Payload Type** as JSON. Save the publisher record and you will be ready to receive data from the GAGEtrak client to the broker. The following sample projects can be used to test the MQTT Publisher capability in GAGEtrak Lite/Pro editions. Before you begin, you will also need to download the Sample Project files from https://gagetrak.com/support.

Using Sample Projects

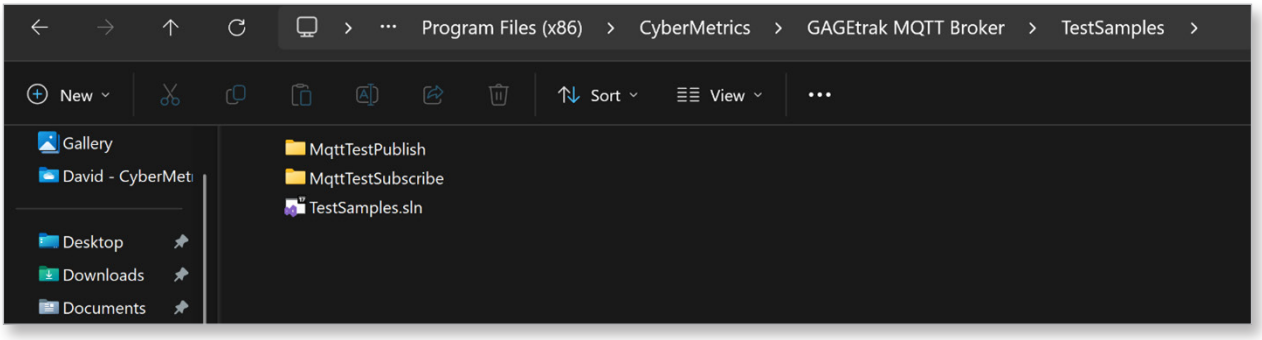
To create a sample project, you will need a code editor, like Visual Studio or Visual Code, etc. You will also need an installed GAGEtrak client with an **Instance Name** and MQTT Publisher record created in **Settings**. Verify that your broker is online and that the client is not blocked by firewall, etc. You can use the starting examples provided to test.

The **MqttSamples** folder contains three projects:

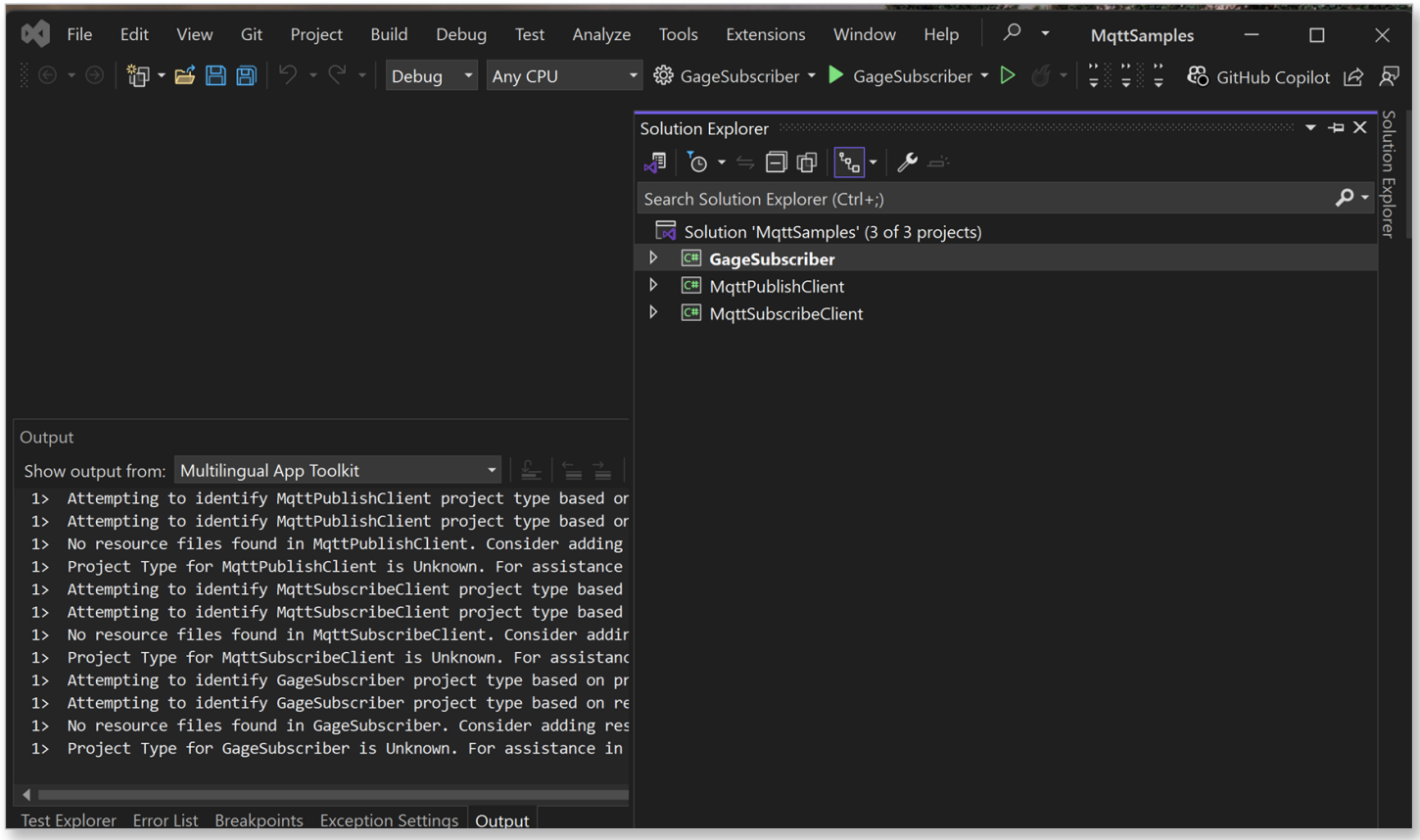


MqttPublishClient and **MqttSubscribeClient** are used to test your broker; they publish and subscribe to a room temperature (these are optional). We will use the project with the **GageSubscriber** to test GAGEtrak MQTT in your environment.

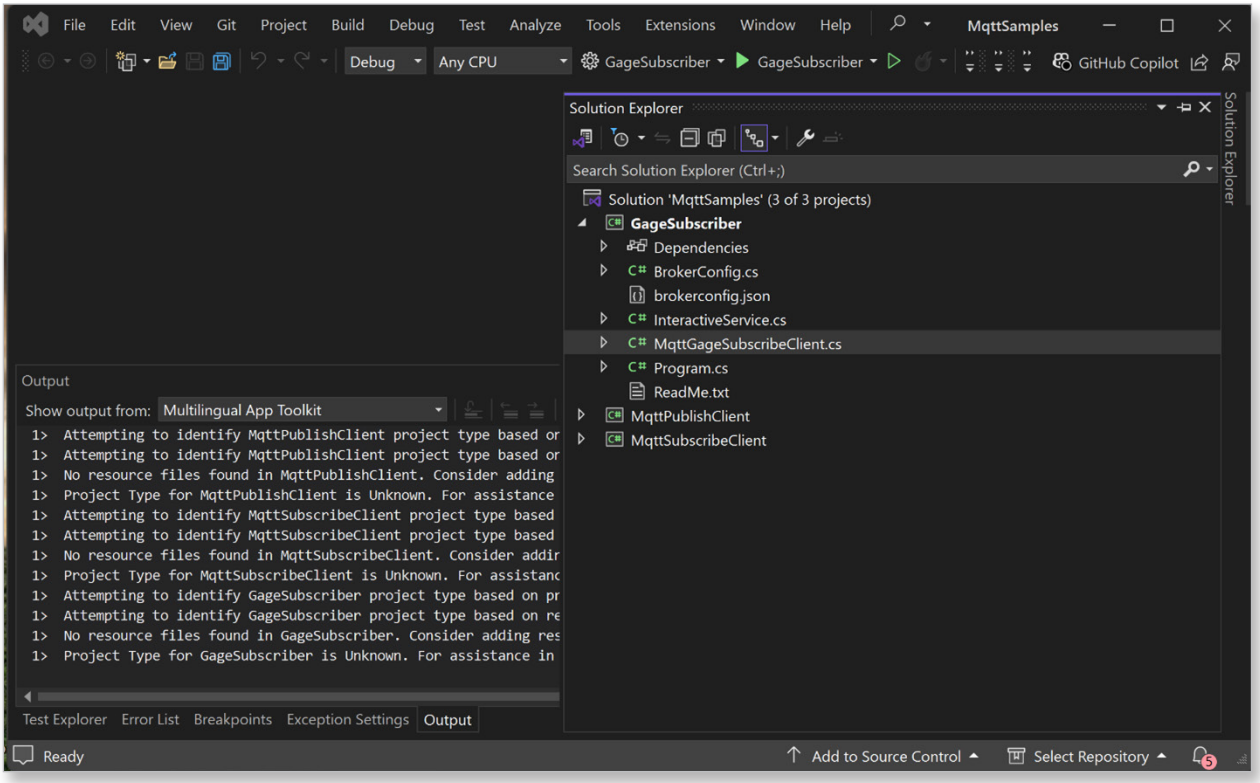
The **TestSamples** directory contains a general publisher and subscriber that is used to do high-speed testing of the broker.



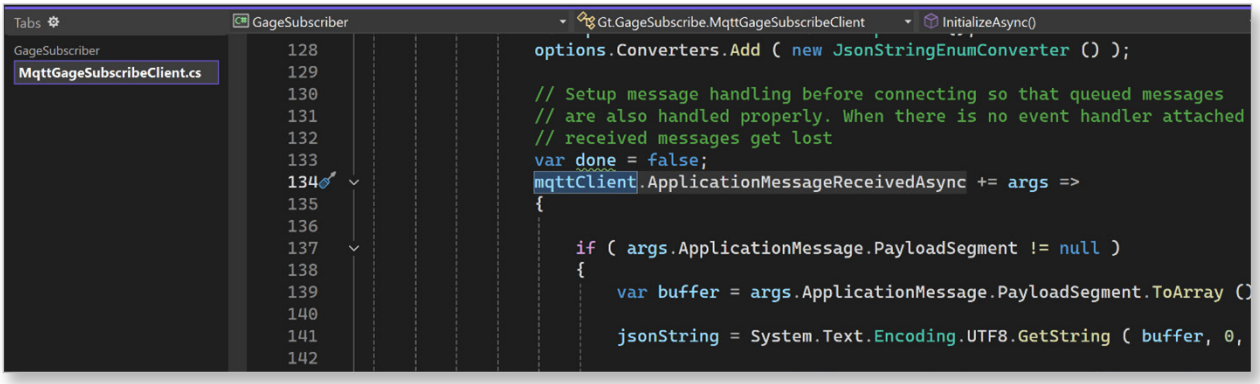
When you are ready to begin testing, go to the installation directory for the broker and in the **MqttSamples** folder, open **MqttSamples.sln** and set **GageSubscriber** as the start-up project.



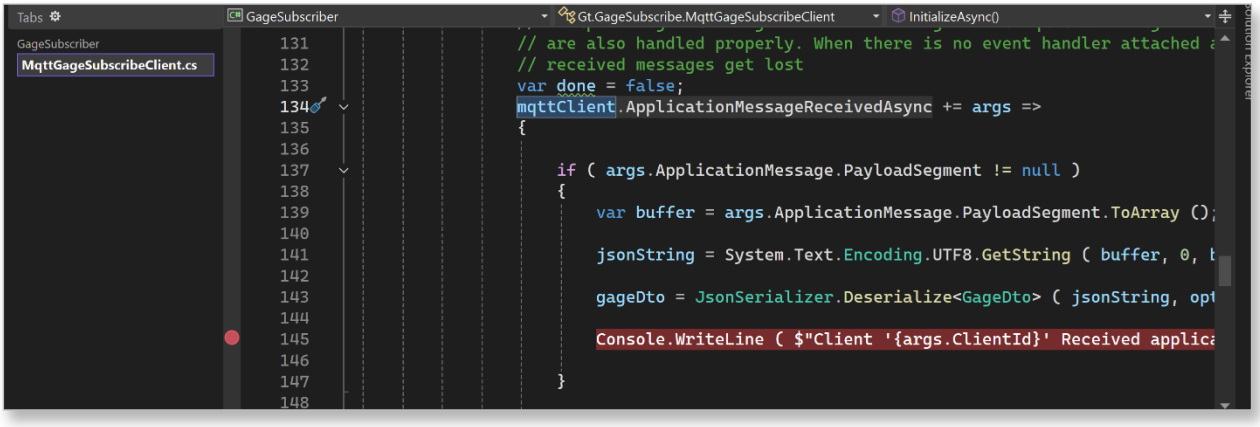
Open the file **MqttGageSubcriberClient.cs**.



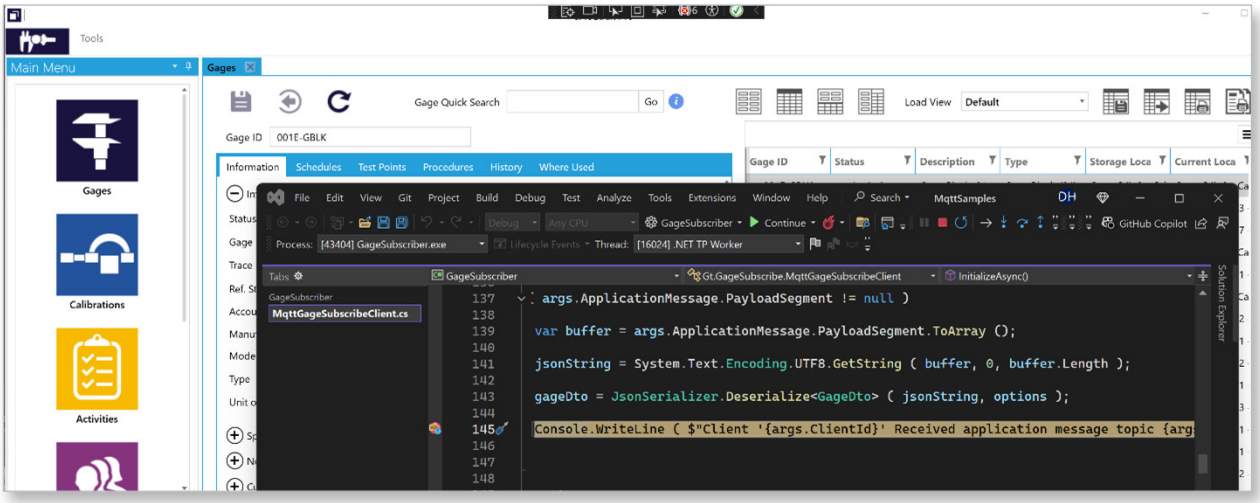
Search for ***mqttClient.ApplicationMessageReceivedAsync*** in the code.



Put a break point after the “if” statement.



Run the project and start the GAGEtrak Pro/Lite client application. In **Gages**, locate any gage record, change the **Accounting No.** field value to ***Foo*** and then save. In a few seconds, your break point will be hit.



If you are not seeing the break point or any output, make sure the GAGEtrak Windows service is running on the broker machine. Verify that your IP address is correct and the MQTT port is not blocked by a firewall on either the broker or GAGEtrak client.

Below is a sample of data changes published from the Sample Project:

MQTT PUBLISHER JSON OUTPUT EXAMPLE

```
{
  "Item_RID": 1,
  "ItemParent": null,
  "ItemTypeName": "Gage",
  "IsChanged": true,
  "IsNew": false,
  "Gage_RID": 1,
  "Gage_ID": "01E-GBLK",
  "Gage_SN": "00-23-172-4562",
  "Asset_No": "Foo",
  "Model_No": "96A6Z",
  "Manufacturer": "Starrett",
  "Drawing_No": null,
  "Drawing_Date": null,
  "Change_Level": null,
  "Change_Date": null,
  "Service_Date": "2024-10-14T00:00:00",
  "Retirement_Date": null,
  "CyclesPerDay": 0,
  "Resolution": null,
  "Operating_Range": "0.100 - 4.0 inch.",
  "Plus_Tolerance": "8",
  "Minus_Tolerance": "8",
  "PassText": null,
  "FailText": null,
  "GM_Cost": 45,
  "Purchase_Date": null,
  "Notes": null,
  "RefStandard": true,
  "GM_Format": null,
  "GM_Usage": 0,
  "AvailableForTransfer": true,
  "TransferCondition": "Complete",
  "CurrentLocked": false,
  "Uncert": 8E-06,
  "Nist_No": "821/123456-37",
  "CustomerInfo": null,
  "RestrictIssue": 0,
  "EstNext_Due_Date": false,
  "AllowIssueWhenPastDue": null,
  "AllowIssueWhenPastDueInKit": null,
  "Custom1": null,
  "Custom2": null,
  "Custom3": null,
  "Custom6": null,
  "StorageLocation_RID_FK": 17,
  "CurrentLocation_RID_FK": 17,
  "CurrentCustodian_RID_FK": null,
  "CurrentTransfer_RID_FK": null,
  "Supplier_RID_FK": 3,
  "Owner_RID_FK": 7,
  "GageType_RID_FK": 17,
  "GageDescription_RID_FK": 15,
  "UnitOfMeas_RID_FK": 1,
  "Status_RID_FK": 1,
  "GageTemplate_RID_FK": null,
  "Custom4_RID_FK": null,
```

Changes triggered by the user action; Changing **Accounting No**. to "Foo" in GAGEtrak (A.K.A. "Asset_No")

From here, you can continue to to test using other changes to gage or calibration records in GAGEtrak to be published to your broker and other subscribers in your network. Otherwise, this concludes the sample project. Please contact technical support if you have any issues.

SUPPORT

+1 (480) 212-1257
1-800-777-7020
support@cybermetrics.com

SALES

+1 (480) 922-7300
1-800-777-7020
sales@cybermetrics.com

GAGEtrak®

GAGETRAK.COM