EBO IoT Edge Smart Connector Extension

INSTALLATION AND CONFIGURATION GUIDE

ECOSTRUXURE BUILDING LABS

Table of Contents

1. Introduction

This document outlines the installation and configuration of the EBO IoT Edge Smart Connector Extension required to integrate Vasakronan's Real Esate Core with EcoStruxure Building Operation. This document assumes that EcoStruxure Building Operations has already been installed and is functional as an independent system.

The following are a list of reference documents:

Document	Description
Smart Connector Installation and Configuration Guide	Complete installation guide for Smart Connector Framework that covers in more depth – installation and configuration options; troubleshooting information on the Smart Connector Framework. This manual will be downloaded during the installation process.
EBO IoT Edge Smart Connector Extension	This manual

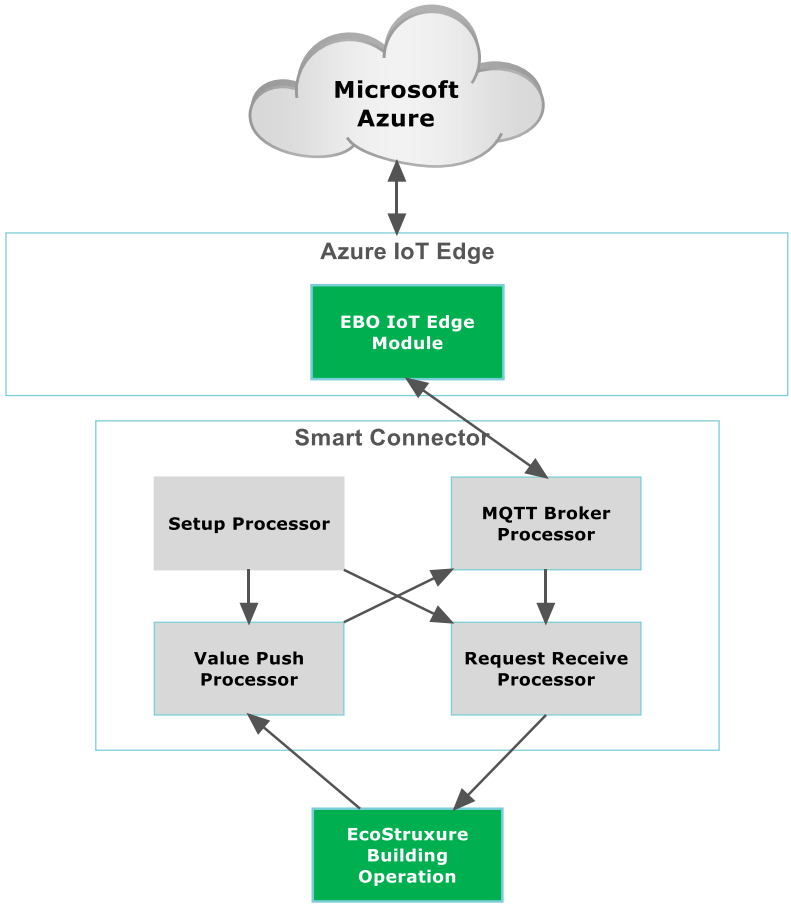
1.1. Versions

This integration has been tested to work with the below versions of the software specified.

Smart Connector: Version 2.4.23 and newer.
EcoStruxure Building Operation: Versions 2.0 and newer
Note: This integration will only be fully functional with EBO versions 2.0 and newer. This is due to performance limitations in older versions of EBO (e.g. SBO 1.8.X and SBO 1.9.X.) However, for testing purposes this integration was made to work with older versions of EBO (e.g. SBO 1.8.X and SBO 1.9.x) in a limited capacity with a maximum of 100 points.

1.2. Architecture

A Basic overview of the architecture can be seen in the below diagram.



How this works in general is that Smart Connector stands up an MQTT Broker, and this MQTT broker brokers messages between the Azure IoT Edge, and EBO. Data is extracted out of EBO using a Smart Connector processor that keeps track of what data needs to be sent, and then formats the data in such a way that the Azure IoT Edge is able to consume it before sending the message to the MQTT broker. At the same time, another Smart Connector processor is always running that will listen for messages from the MQTT Broker so that data sent from the Azure IoT Edge can be writen into EBO.

1.3. Prerequisites

In order to install the EBO IoT Edge Smart Connector Extension, we must first install and license the Smart Connector Framework. There are multiple configuration options as to where the Smart Connector Framework can be installed – for use in this document; the Smart Connector Framework and Extension will be installed on the same machine as the EcoStruxure Enterprise Server and SQL Express. For additional options using SQL or remote servers not containing the Enterprise Server refer to the Smart Connector Installation and Configuration Guide.

The following prerequisites must be performed before you start the installation and configuration of the Smart Connector Framework and EBO IoT Edge Smart Connector Extension.

- EcoStruxure Building Operation Enterprise Server - Installed, Configured and Functional
- Microsoft .NET v4.5 or later must be installed on the Enterprise Server
- Install all Windows updates on the Enterprise Server
- SQL Express is installed on the Enterprise Server (or server for Smart Connector installation
  Note: if SQL is installed on a remote machine follow the detailed instructions in the Smart Connector Framework Installation and Configuration Guide.pdf
- The specified user must have at least the public and dbcreator user roles in the SQL server

Note: Additional Installation options for installing the Smart Connector Framework can be located in the Smart Connector Installation and Configuration Guide.

1.4. Quick Start Installation Sequence

The following overview provides the steps necessary to install and configure the system. The subsequent chapters will provide detailed information for each step in the process.

1. Install, License, and Configure the ***Smart Connector Framework***
2. Install, License and Configure the ***EBO IoT Edge Smart Connector Extension***

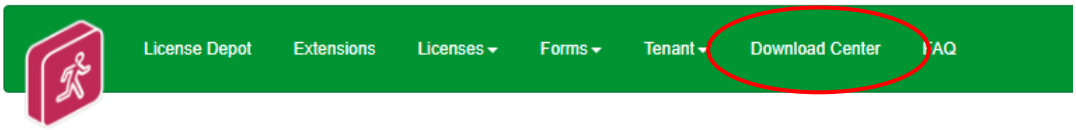
2. Smart Connector Framework Installation

The first step in the process is to download Smart Connector Framework software from www.smartconnectorserver.com, once downloaded you will install the Smart Connector Framework software, obtain the machine thumbprint, license the Framework to the machine thumbprint and finally configure the Framework system. Once the Smart Connector Framework has been installed, configured and licensed we can extend the Framework by adding the EBO IoT Edge Smart Connector Extension.

2.1. Downloading the Smart Connector Framework

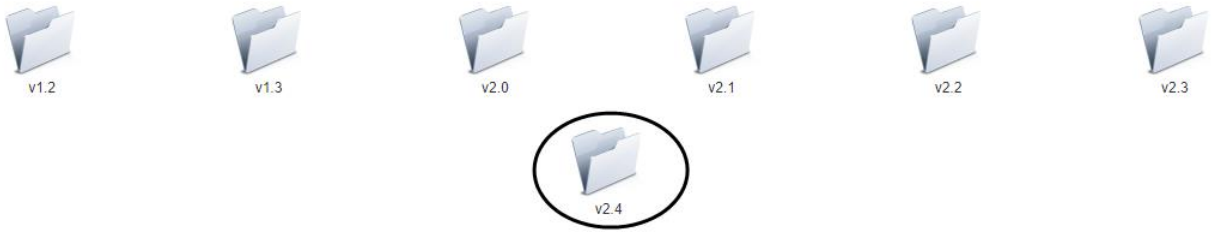
The following steps will assist in downloading the Smart Connector Server Framework.

1. Go to www.smartconnectorserver.com
2. Request credentials to logon to the web site
3. Log on to the web site
4. From the menu, select Download Center from the menu

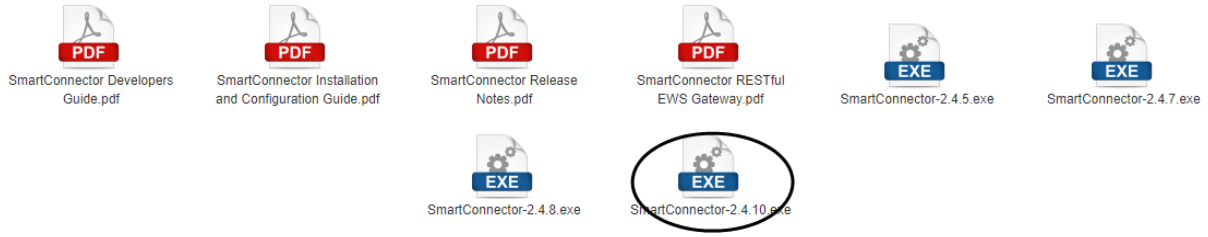


5. Select Smart Connector icon

6. Select the latest folder version (v2.4)



7. Select the latest version of Smart Connector (v2.4 as of writing this document)
   (Note: make sure Popups are not blocked by your browser)



8. Save the latest Smart Connector version download file
   ***Note: This document will be using* v2.4.10.exe as that was the latest available during the writing of this document, but the latest version should be used.**
9. Select the Smart Connector Installation and Configuration Guide.pdf
10. Save the Smart Connector Installation and Configuration Guide.pdf download file

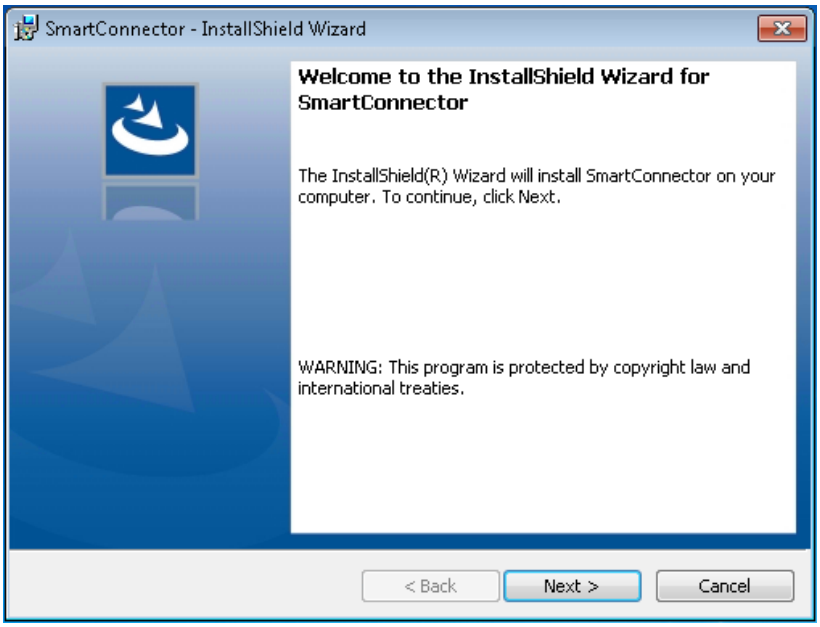
2.2. Installing the Smart Connector Framework

To install the Smart Connector Framework, execute the setup file that was just downloaded. Run SmartConnector-2.4.23.exe – You must run this as an *Administrator*.

- Install Smart Connector Framework
- Validate Smart Connector Framework Installation
- Change Default Credentials
- Install Smart Connector Framework Runtime License

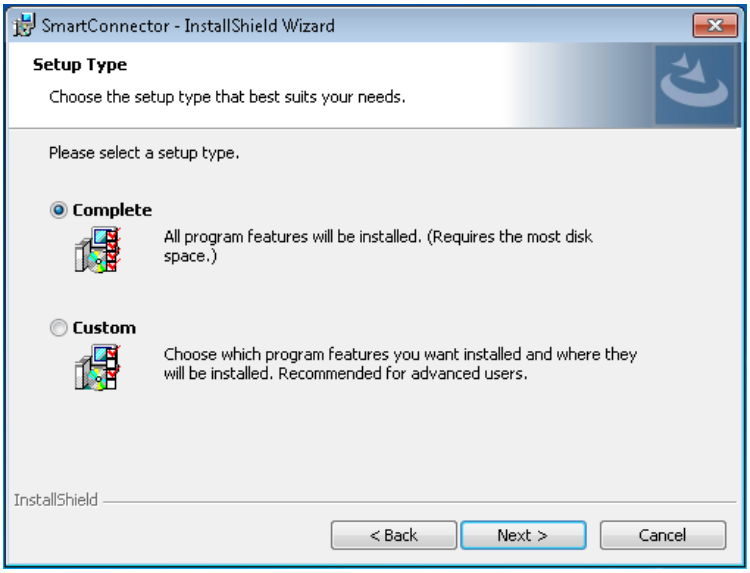
2.2.1. Installing Smart Connector Framework

1. Locate the downloaded file SmartConnector-2.4.23.exe
2. Right click on the file SmartConnector-2.4.23.exe
3. Select Run as Administrator



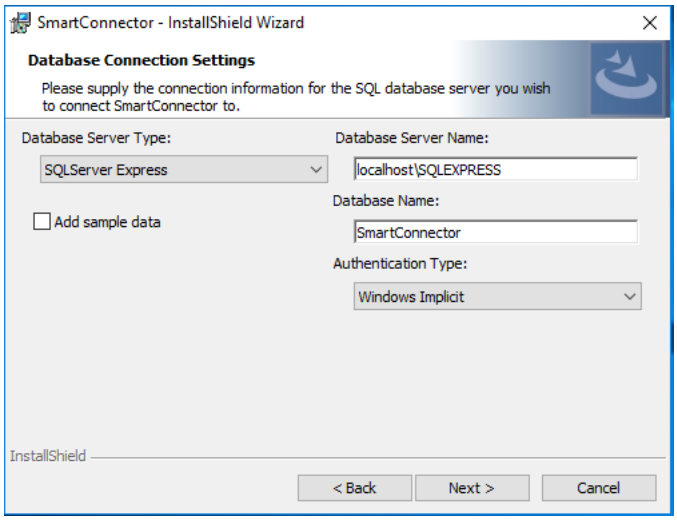
4. Click **Next**.
5. Review and accept the terms to the End User License Agreement

6. Click **Next**.



7. Choose the Setup Type you wish to perform. If this is a new installation, ***you must choose Complete***.
8. Click **Next**.

9. Enter the required information for the database server where you will install the database to. *(Note: for this manual example we are using SQL express and a local Windows user)*



i) You can **uncheck** - Add sample data

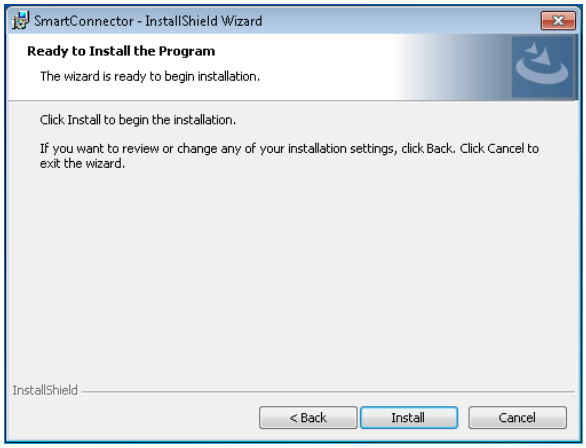
ii) Select the Database Server Type: **SQLServer Express**

iii) Select the Authentication Type: **Windows Implicit**

Note: The logged in user must have at least the public and dbcreator user roles in the local SQL server. In this configuration Smart Connector runs under the NT Authority\System account. *Appendix B*

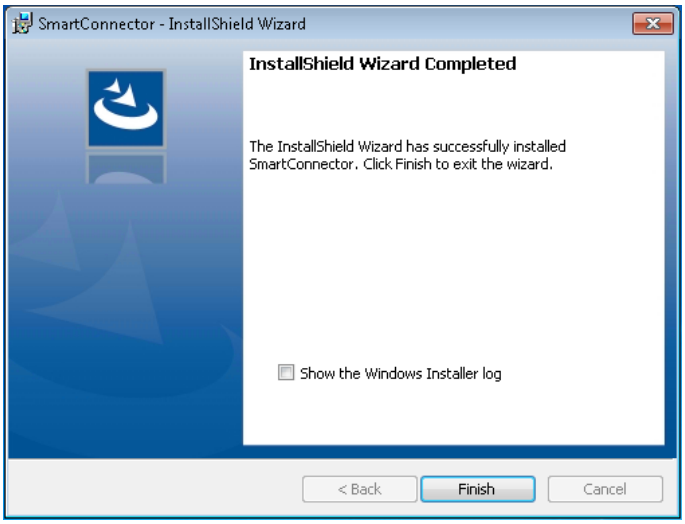
For additional SQL installation options, refer to the Smart Connector Installation and Configuration Guide previously downloaded

10. Click **Next** to display the final confirmation dialog shown below.



11. Click **Install** to complete the installation and create the default database.

12. Click **Finish**.

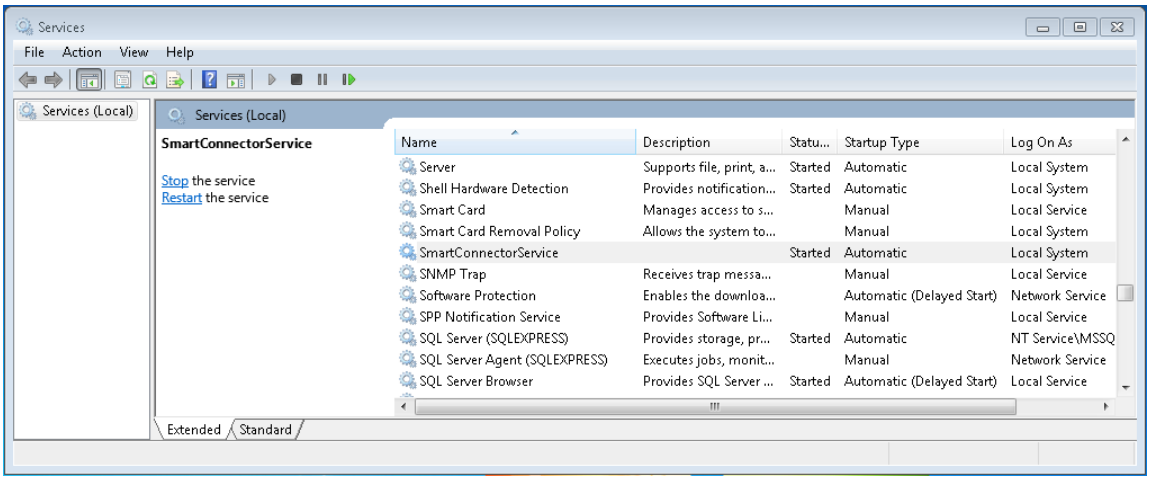


2.2.2. Validating Smart Connector Framework Installation

To review the service installation, you should perform the following:

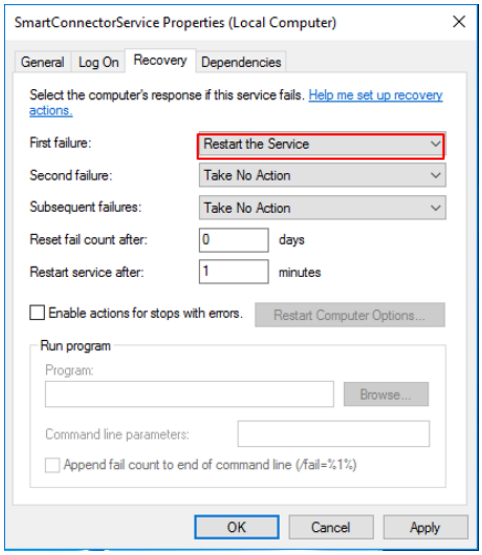
1. Open the Windows Services dialog.
2. Find the entry for "SmartConnectorService". It should have a Status of "Started" or "Running" and a Startup Type of "Automatic" as shown below.

   If Smart Connector and the connected database server are located on the same physical server, we recommend changing that the Startup Type to "Automatic (Delayed Start)".



3. Right click the "SmartConnectorService" entry and choose Properties.
4. Click the **General Tab**.
5. Confirm the Startup Type is **Automatic**.
6. Click the **Log On** tab.
7. Confirm that the "Local System account" is selected. This may be different depending on the database authentication type you chose earlier.
8. Click the Recovery tab.

9. Set First failure: to **Restart the Service**
   We recommended that you choose at least one recovery action in the event that the Smart Connector Service experiences a failure. At a minimum, “Restart the Service” should be selected



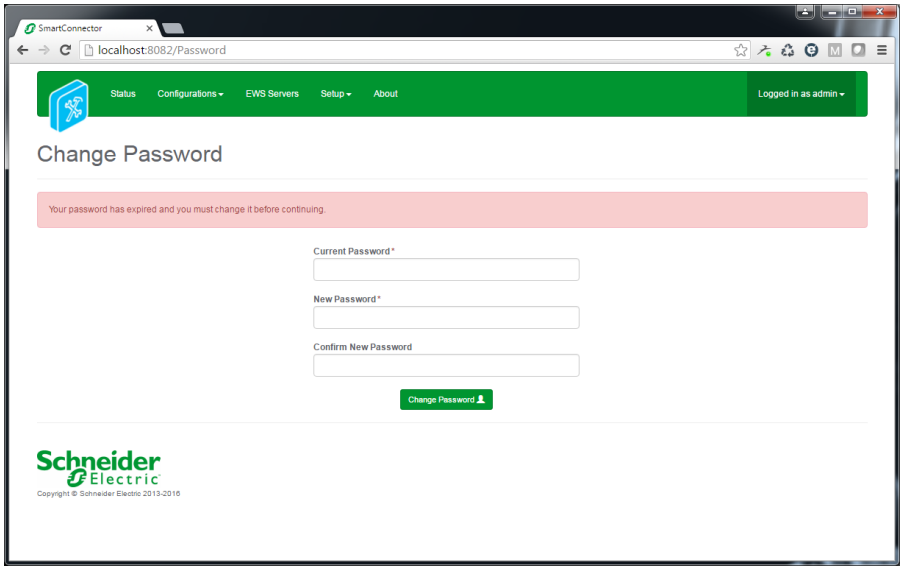
10. Select **OK** to save all changes

2.2.3. Changing Default Credentials

By default, Smart Connector will enable Smart Connector Portal on the local machine. Using Smart Connector Portal, you must change the default password to a new password.

1. Open a web browser
2. Navigate to http://localhost:8082
3. At the Login Page, enter the default user credentials of admin and Admin!23.

   At this point you will be presented with the Change Password Page as show below.



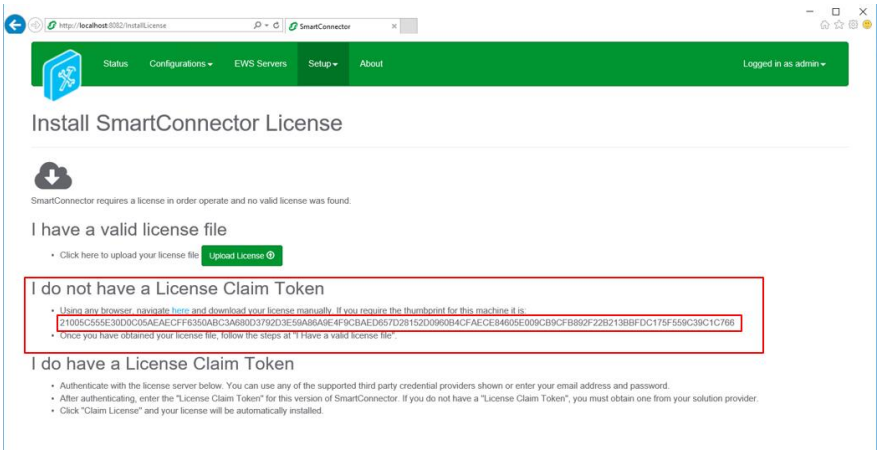
4. Enter the default password as the Current Password.
5. Enter a new password. Portal passwords are required to be at least 6 characters in length and contain a mix of upper case, lower case, numeric, and at least one non-alphanumeric character.
6. Confirm the password you entered in step 5.
7. Click **Change Password**.
8. Re-authenticate (Login) with your User name and New password.

2.2.4. Installing the Smart Connector Framework Runtime License

Smart Connector Framework requires a license in order to run. After changing the default password, navigating to any page of Smart Connector Portal will return the user to the Install License page where a runtime license must be installed.

Smart Connector Connected to the Web

If the Windows machine with Smart Connector Framework detects an active internet connection, the Install Smart Connector License page will automatically be displayed. Once authenticated with the License Manager, you only need to enter a License Claim Token to "claim" the runtime license and it will be automatically installed. Alternatively, the user may click "Upload License" to manually upload an already obtained license file. License Claim tokens and license files can be obtained from www.smartconnectorserver.com.



Smart Connector Not Connected to the Web

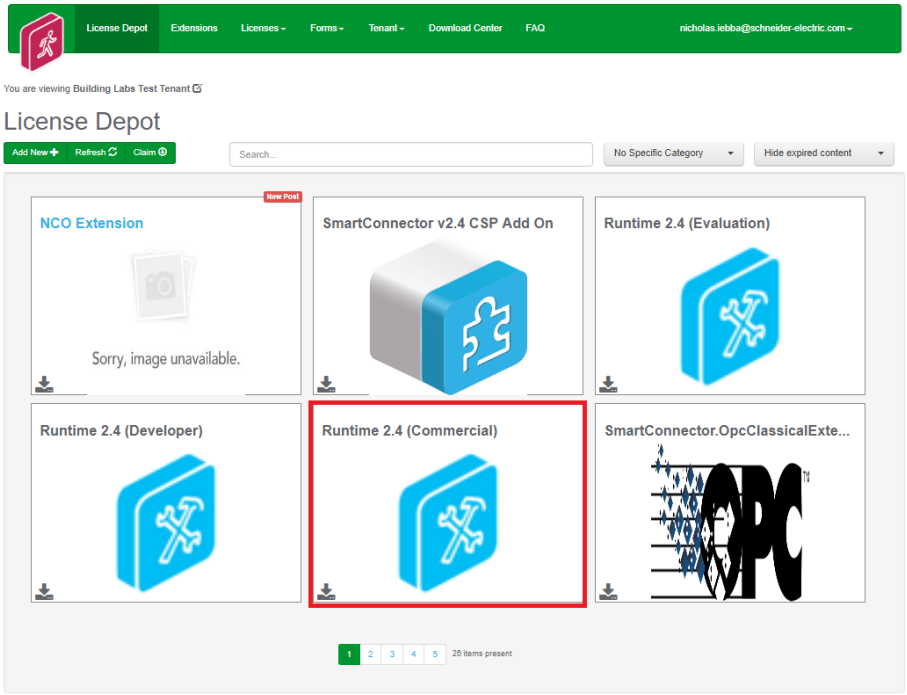
If Smart Connector fails to detect an active internet connection, the Install License page shown below will be displayed.

Directions are provided on how to download a license file from www.smartconnectorserver.com.

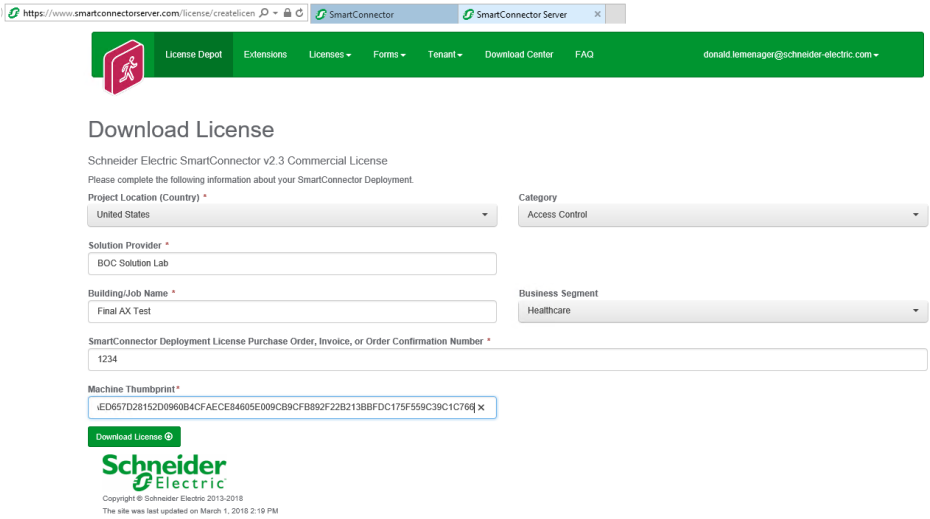
Obtain a license when you do not have a Claim Token

If you do not have a claim token then you can download a License for Smart Connector Framework via a file and the Thumbprint of the machine smart Connector Framework has been installed on.

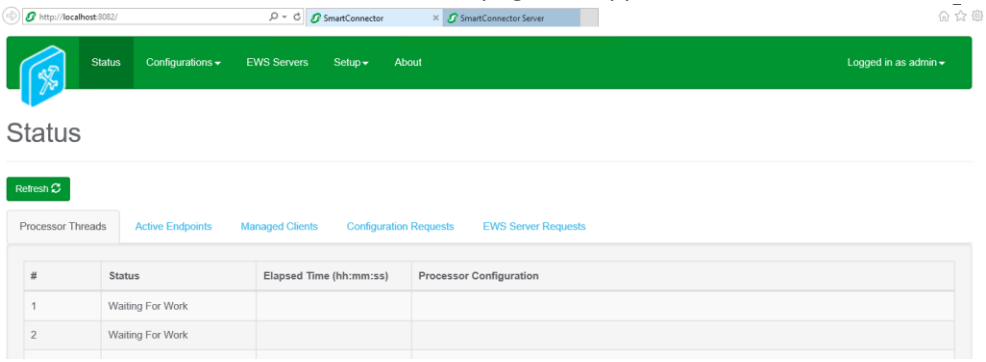
1. From the "I do not have a License Claim Token section of the Smart Connector License page"
2. Copy the Machine Thumbprint into the Windows clipboard for use later
3. Click on the navigate here button in this section, this will connect you to the License Depot web page
4. Log on to the License depot web page with your smartconnectorserver.com credentials
5. Scroll down until you see the Runtime v2.4 commercial license



6. Select the ( ) download button to obtain the License file
7. Complete the Download License form



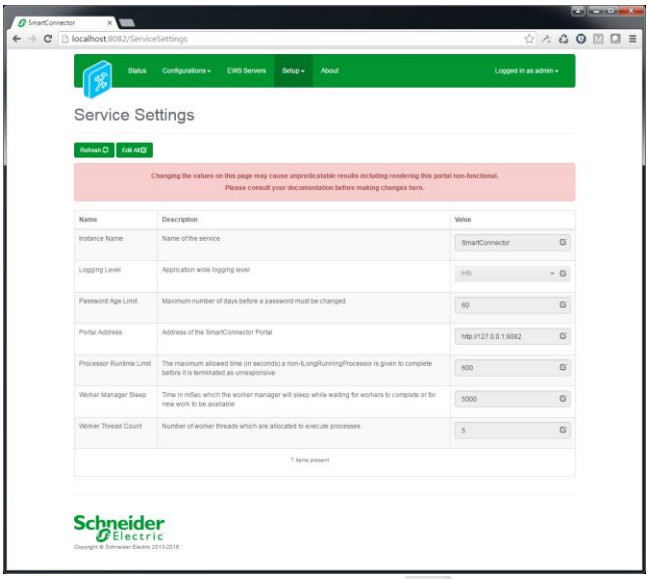
8. Paste in the machine thumbprint from the Windows clipboard (copied earlier)
9. Save the downloaded License file
10. Return to the Install Smart Connector License page
11. Select Upload License
12. Smart Connector Framework is now successfully licensed
13. The Smart Connector Framework status page will appear



Confirm Settings

Smart Connector installs the service with some default settings. After changing the password, you should confirm the system settings meet the criteria for how Smart Connector Framework will be used.

1. Open any web browser
2. Navigate to http://localhost:8082
3. Authenticate with the credentials you used in the prior section.
4. From the menu, click **Setup -> Service Settings**.



To edit any field, you can either click the edit icon ( ) in that field or click the Edit All button to enable all fields for editing.

The default settings will be acceptable for the initial installation of Smart Connector Framework.

Users should use good security practices to define the expiration time for user Passwords.
The EWS Portal address can also be modified here from the default port used 8082.

5. Review and/or change values as desired. Unless otherwise noted, changes made here will take effect without a service restart.

Instance Name	Appears in the browser tab and can be useful to distinguish which Smart Connector instance you are looking at if you are connecting to multiple deployed instances from a single browser.
Logging Level	Maximum level Smart Connector will log. Possible values are *None, Error, Status, Info, Debug, Trace, All*. This setting is used in conjunction with Logging Filters to control how much information is captured in the log files.
Password Age Limit	The maximum number of days before a Portal user's password will expire.
Portal Address	Address of Smart Connector Portal. For security concerns, the default value will be 127.0.0.1 which means the portal can only be accessed from the local machine. If broader access is required, this value can be modified by using the "+ syntax" e.g. http://+:8082. This will allow access to any IP or DNS which resolves to the local machine. If you plan to secure the endpoint with a certificate, then the protocol shown here should be changed to https to match. Entering an empty value will disable the portal. Use caution! Consult the Security Considerations for suggestions on how best to configure this.
Processor Runtime Limit	The maximum amount of time a Processor Configuration is given to complete before it is deemed to be unresponsive and is terminated. Unless otherwise instructed this value should not need to be modified.
Worker Manager Sleep	The amount of time that the Worker Manager will idle before determining if there are Processors that need to be invoked. Unless otherwise instructed this value should not need to be modified.
Worker Thread	**Count** – The number of concurrent Processors that can be executed. This number may be increased but is largely dependent on the host machine's number of logical processors. To determine the number of logical processors, open a command prompt and enter the command: `WMIC CPU Get DeviceID,NumberOfCores,NumberOfLogicalProcessors`. While you can set this value greater than the number of logical processors, it represents the number of concurrent workers that can run without potential operating system queuing. You will need to restart the Smart Connector Service for this change to take effect.

6. After you have made the necessary changes, click Save to save them to the database.

3. EBO IoT Edge Smart Connector Extension Installation

The steps below will walk you through how to obtain, license, and configured the EBO IoT Edge Smart Connector Extension

3.1. Obtaining the EBO IoT Edge Smart Connector Extension

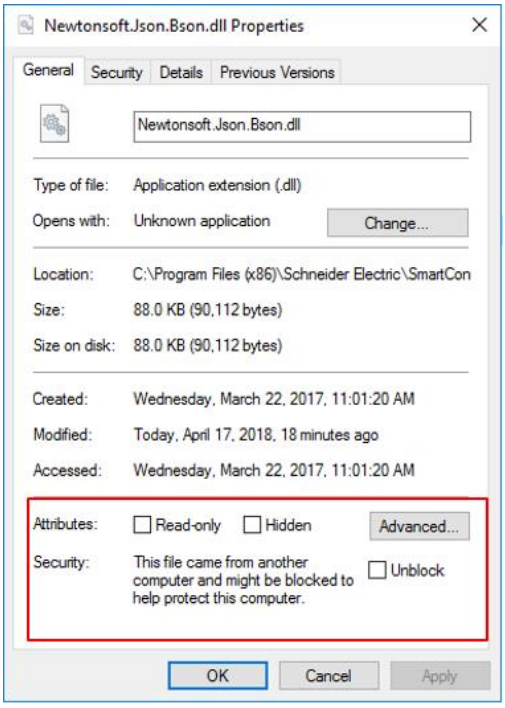
Because this extension was not developed for broad distribution, this extension is not publically available. Please email TODO@schneider-electric.com to obtain the files required for this Smart Connector Extension.

3.2. Installing the EBO IoT Edge Smart Connector Extension

Because this extension was not developed for broad distribution, this extension is not publically available. Please email TODO@schneider-electric.com to obtain the files required for this Smart Connector Extension.

1. Extract the files from the zip file to a temporary directory
2. Right click on each file and select Properties
3. Verify the file is not blocked – see screen shot below;

a. If the file is blocked, select Unblock



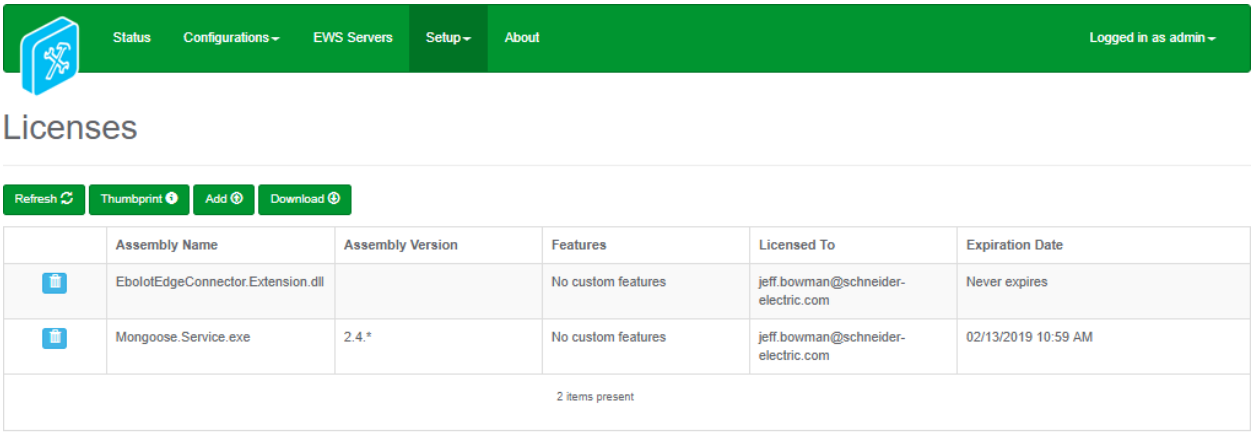
4. Copy the files to the installed directory for Smart Connector Framework (e.g. C:\Program Files (x86)\Schneider Electric\SmartConnector)
5. You have successfully installed the EBO IoT Edge Smart Connector Extension

3.3. Licensing the EBO IoT Edge Smart Connector Extension

The license can be obtained via the claim token that should have been emailed when this extension was downloaded from https://shop.exchange.se.com. If you did not download this extension from the exchange and require a license, please contact support.seal@se.com.

Once you have obtained the claim token, follow the below steps to add this license to your Smart Connector installation.

1. Go to the smartconnector portal
2. Select setup -> **License**
3. Select **Download**
4. Enter your claim token.



5. If you are not able to claim your token from the Smart Connector portal, please visit www.SmartConnectorServer.com, click on the License Depot link, and click the claim button. The download license file can be uploaded to the Smart Connector portal by clicking on the **Add+** button instead of **Download.**
6. You have successfully licensed the EBO IoT Edge Smart Connector Extension.

3.4. Configuring the EBO IoT Edge Smart Connector Extension

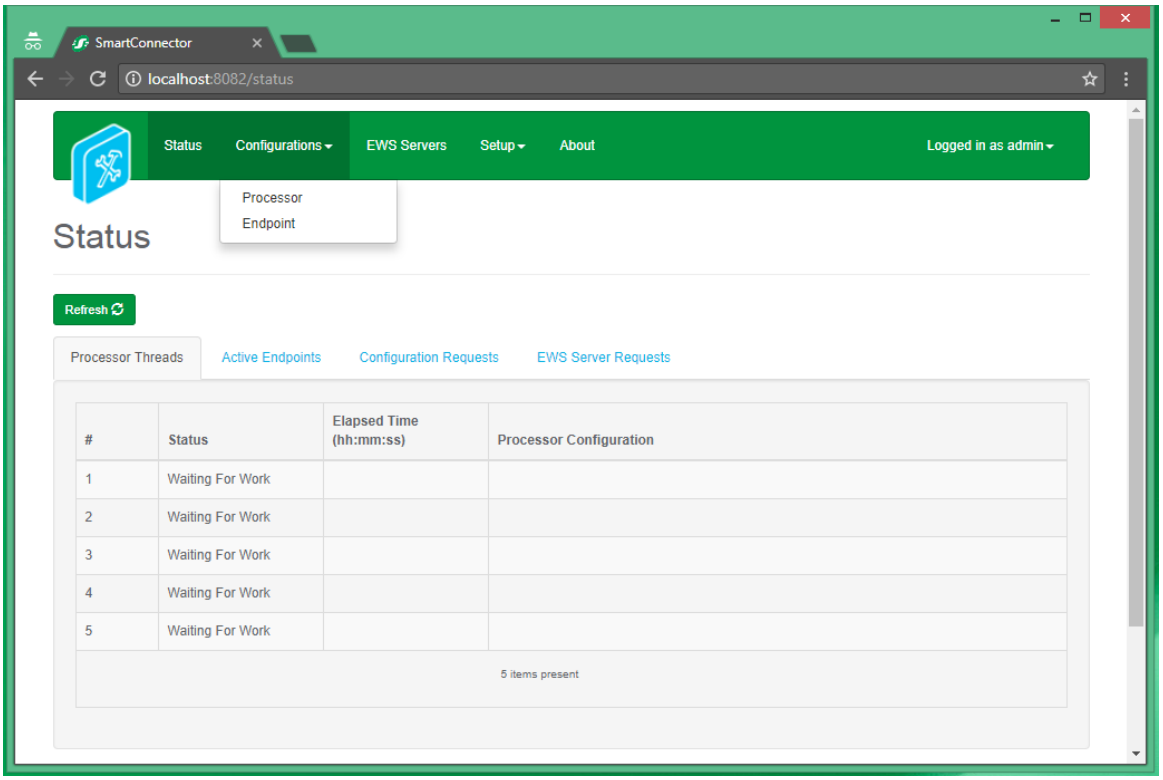
The EBO IoT Edge Smart Connector extension consists of four separate processors, that when configured will work together to discover points, get updated values and push that data to the IoT Edge, and received requests from the IoT Edge for writing values back into EBO.

The four processors are:

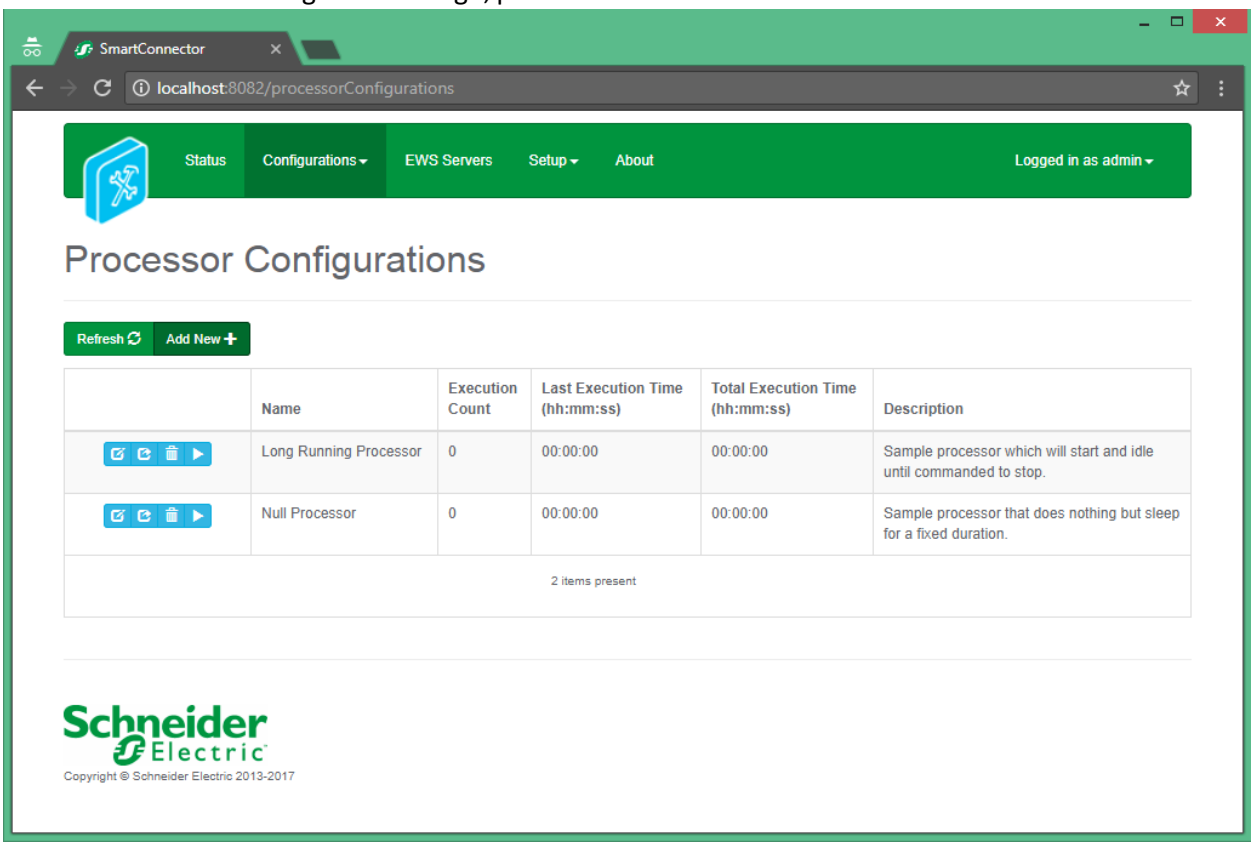
- MQTT Broker Processor: This processor stands up an MQTT broker, which will broker requests between Smart Connector and the IoT Edge.
- Setup Processor: This processor parses the signals CSV file, and stores the result in the in-memory cache for use by the other processors.
- Value Push Processor: This processor gets runtime values from EBO, and pushes them to Azure as defined by the signals CSV file.
- Request Receive Processor: This processor runs forever and manages requests from the IoT Edge for setting values back into EBO.

The following steps are the common steps that must be followed for each of the four processors when configuring them.

1. Log into the **Smart Connector Portal**. If Smart Connector is installed on the same machine use localhost:8082
2. Select **Configurations -> Processor**.

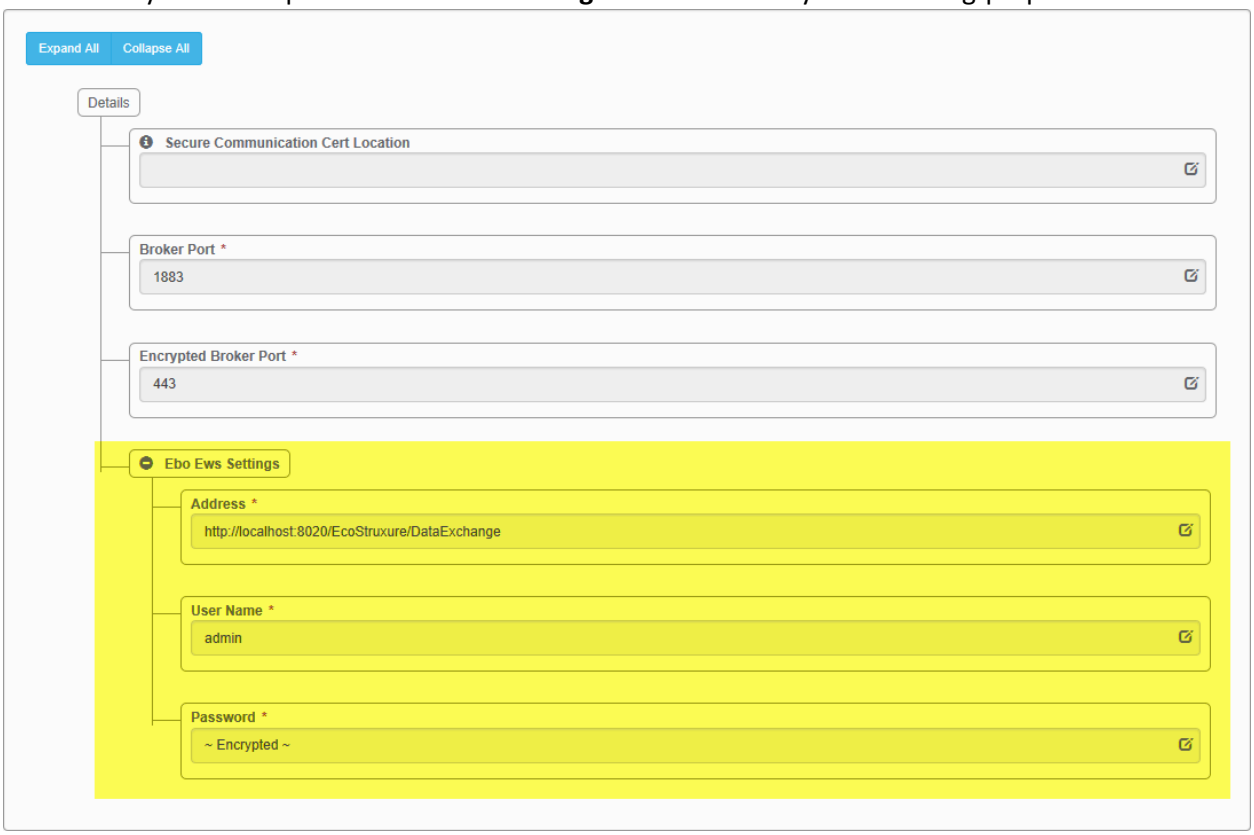


3. From the Processor Configurations Page, press Add New + Button.



4. From the **Add Processor Configuration** Page, Select the **EboIotEdgeConnector.Extension** Assembly
5. Select the **Next** Button.
6. Choose the **EboIotEdgeConnector.Extension.{ProcessorName}** class and press the **Next** Button.
7. Give a Name and a Description for this configuration and Press the **Finish** Button.
8. On the **Process Configuration** Page, Click on the **Details** Tab.

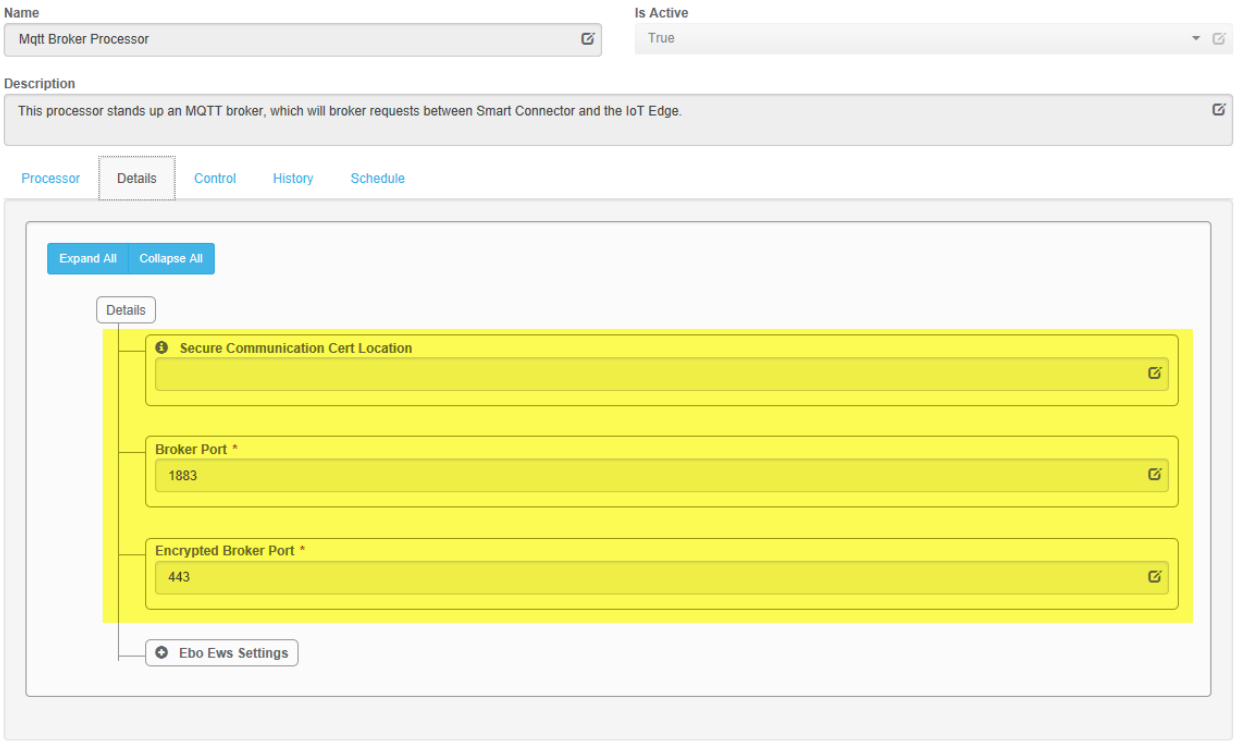
9. Click the + symbol to expand the **Ebo Ews Settings** Node and modify the following properties:



- Address: The EWS Endpoint for the EBO server you will be connecting in the following format ***http://{serverIp}:{serverPort}/EcoStruxure/DataExchange***
- User Name: The EBO user that will be used to connect to EBO. **Default:** admin
  Note: The EWS user must be in the default EBO domain.
- Password: The password for the user entered as the User Name.

3.4.1. MQTT Broker Processor

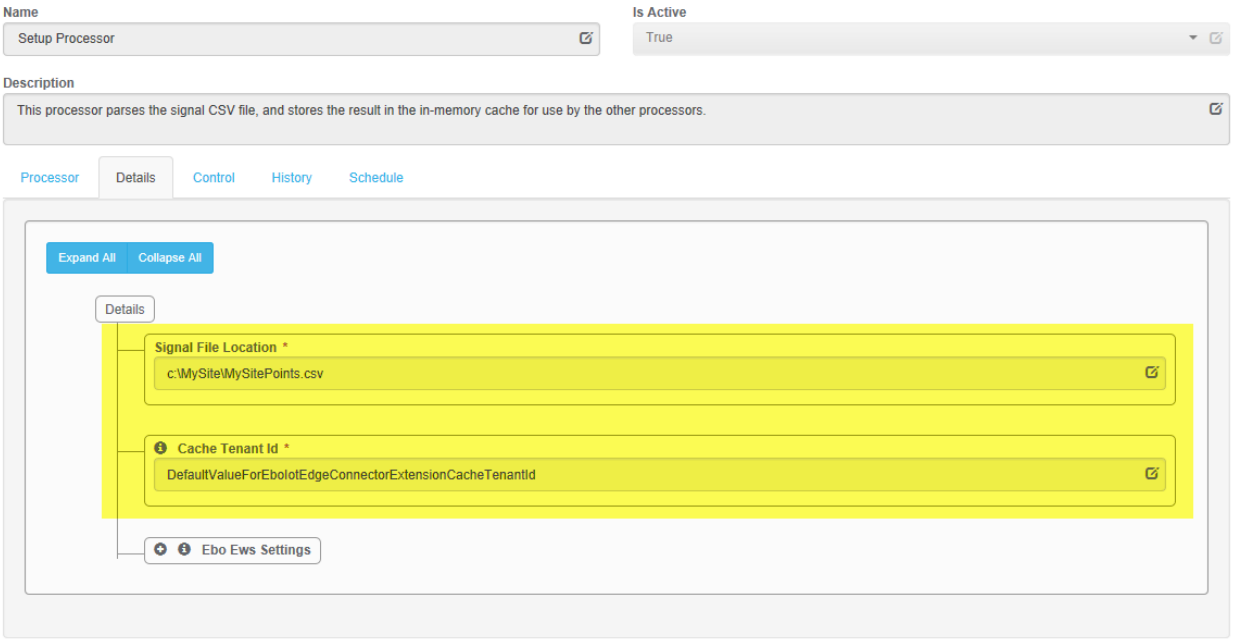
The below steps detail the specific settings that need to be configured for the MQTT Broker Processor



- Secure Communication Cert Location: The location on your PC where your '.cer' certificate file is stored. If this is left blank, non-secure communication will be used. **Default:** *empty.*
- Broker Port: The port used by the broker when the broker is not using secure communication. **Default:** 1883
- Encrypted Broker Port: The port used by the broker when the broker is using secure communication. **Default:** 443.

3.4.2. Setup Processor

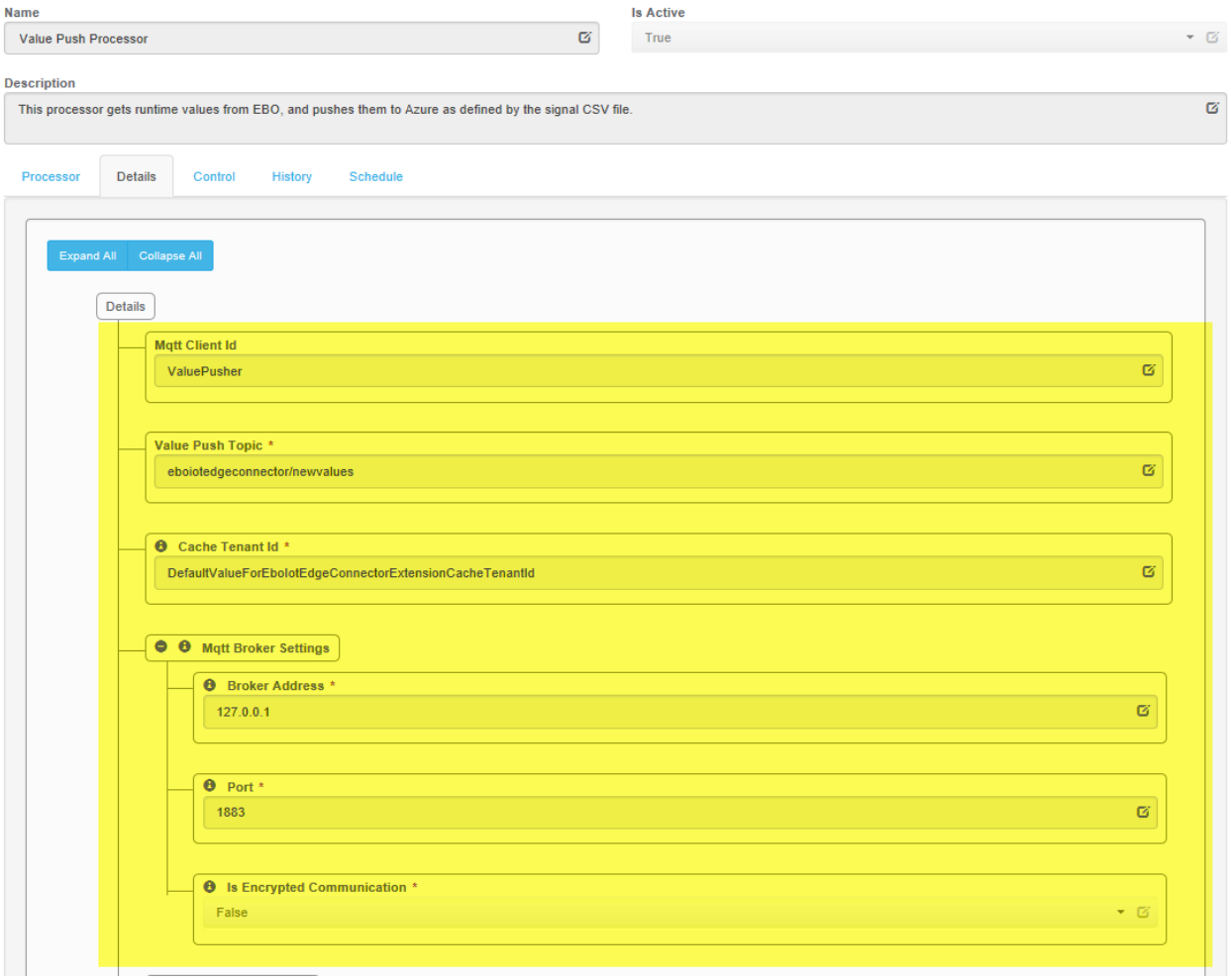
The below steps detail the specific settings that need to be configured for the Setup Processor.



- Signal File Location: The location on your PC where the semicolon separate .csv that contains all the points is located. (this file should be obtained from your system integrator.)
- Cache Tenant Id: The cache tenant ID is what is used internally by Smart Connector so that the other processors can share data. The default value for this should not be changed unless specifically told to do so. **Default:** *DefaultValueForEboIotEdgeConnectorExtensionCacheTenantId*

3.4.3. Value Push Processor

The below steps detail the specific settings that need to be configured for the Value Push Processor

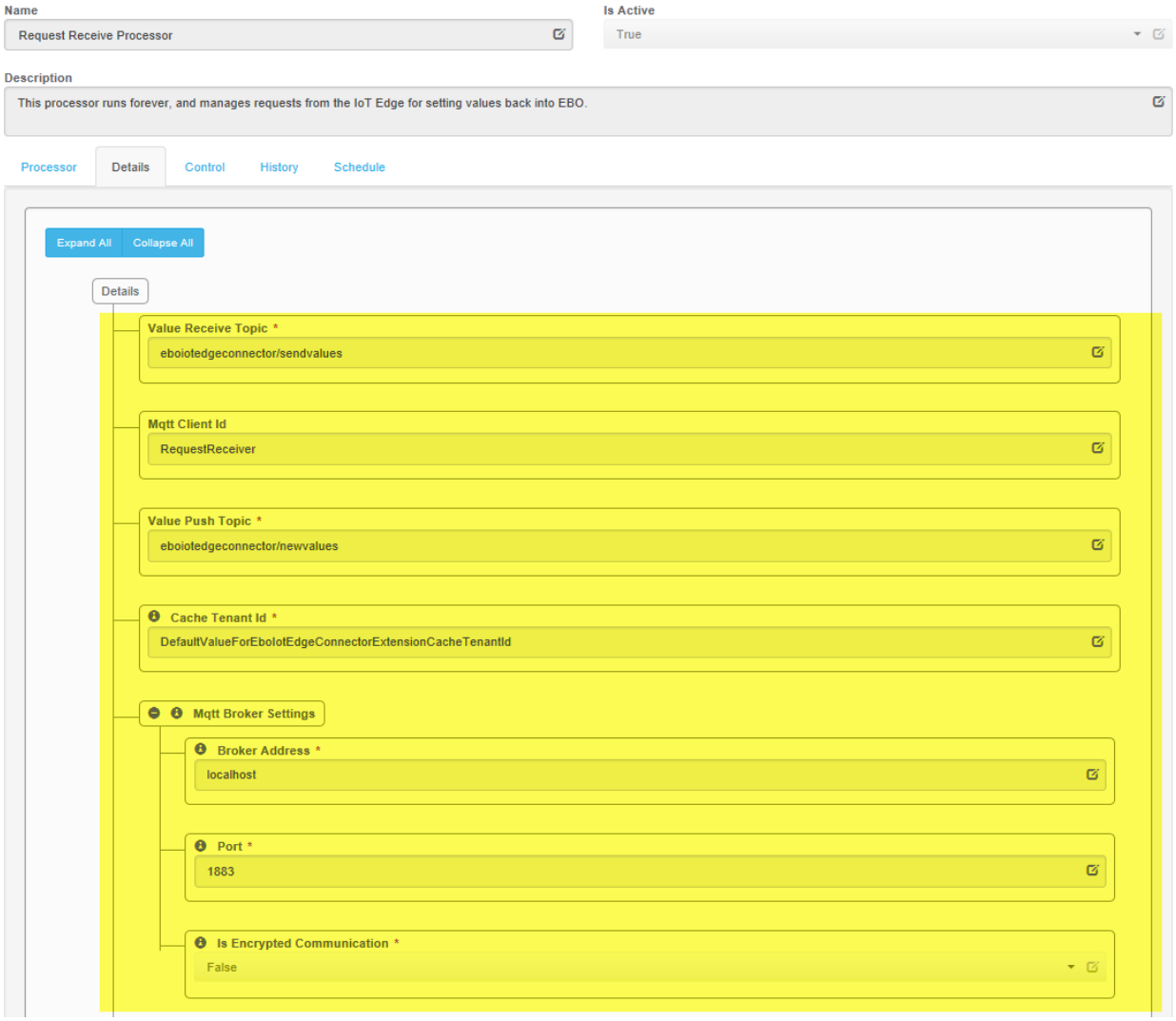


- Mqtt Client Id: The ID that identifies the Value Push Processor to the MQTT Broker when it connects. This should be unique. **Example:** ValuePusher
- Value Push Topic: This is the MQTT topic that the updated values will be pushed to. The default value for this should not be changed unless specficially told to do so. **Default:** *eboiotedgeconnector/newvalues*
- Cache Tenant Id: Cache Tenant Id: The cache tenant ID is what is used internally by Smart Connector so that the other processors can share data. The default value for this should not be changed unless specifically told to do so. **Default:** *DefaultValueForEboIotEdgeConnectorExtensionCacheTenantId*
- Mqtt Broker Settings: The below settings are used for connecting to the MQTT broker started by the MQTT Broker Processor.
  - Broker Address: The address of the MQTT Broker. **Default:** *127.0.0.1*
  - Port: The port of the MQTT Broker. **Default:** *1883*

- Is Encrypted Communication: Whether or not this MQTT client should secure communication. **Default:** *False*

3.4.4. Request Receive Processor

The below steps detail the specific settings that need to be configured for the Request Receive Processor.



- Value Receive Topic: This is the MQTT topic that the processor will be listening on for values to written to. The default value for this should not be changed unless specficially told to do so. **Default:** *eboiotedgeconnector/sendvalues*
- Mqtt Client Id: The ID that identifies the Value Push Processor to the MQTT Broker when it connects. This should be unique. **Example:** ValuePusher
- Value Push Topic: This is the MQTT topic that the the updated values will be pushed to. The default value for this should not be changed unless specficially told to do so. **Default:** *eboioteedgeconnector/newvalues*
- Cache Tenant Id: Cache Tenant Id: The cache tenant ID is what is used internally by Smart Connector so that the other processors can share data. The default value for this should not be

changed unless specifically told to do so. **Default:** *DefaultValueForEboIotEdgeConnectorExtensionCacheTenantId*

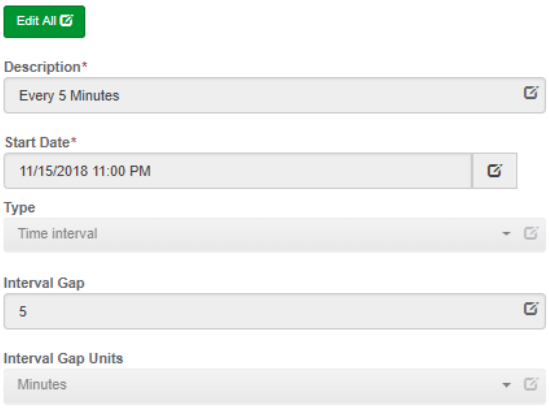
- Mqtt Broker Settings: The below settings are used for connecting to the MQTT broker started by the MQTT Broker Processor.
  - Broker Address: The address of the MQTT Broker. **Default:** *127.0.0.1*
  - Port: The port of the MQTT Broker. **Default:** *1883*
  - Is Encrypted Communication: Whether or not this MQTT client should secure communication. **Default:** *False*

3.5. Configure scheduled execution of Processors

The following procedures will create schedules that will be used to control the execution of each processor. These schedules will then be attached to each processor.

3.5.1. Configure a 5 Minute schedule

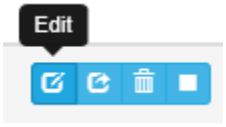
1. Select **Setup -> Configuration Schedules**.
2. From the **Configuration Schedules** page, select **Add New +**
3. Enter *'Every 5 Minutes'* in the **Description** field
4. Select the current date and time from the **Start Date** field
5. Select **Time Interval** from the **Type** field
6. Enter *'5'* in the **Interval Gap** field
7. Select **Minutes** from the **Interval Gap Units** field
8. Select **Save** to save the Schedule

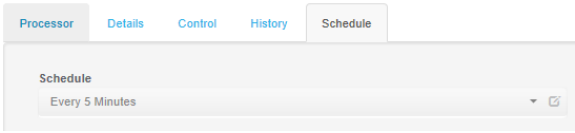


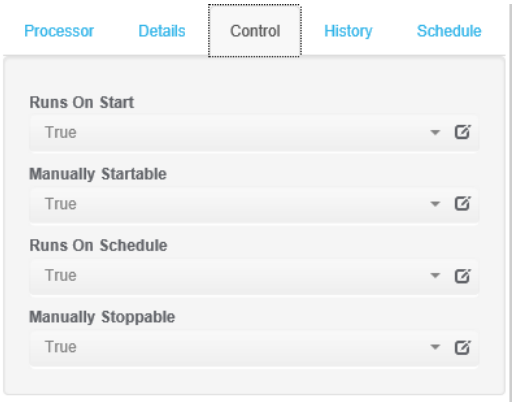
3.5.2. Assign a Schedule to the Processors.

The following processor should be done for the **MQTT Broker Processor**, the **Value Push Processor**, and the **Request Receive Processor**, but ***NOT*** the Setup Processor. Do not assign a schedule for the Setup processor, but do step '6' for it.

1. Select **Configurations -> Processor**.
2. Select **Edit** for one of thre
3. Select the **Schedule** tab
4. Select the **Every 5 Minutes** schedule from the **Schedule** field
5. Select the **Control** tab
6. In the **Runs On Start** field select **True**
7. In the **Runs On Schedule** field select **True**
8. Select **Save** to save the changes to the Processor Configuration







3.5.3. Force the Setup Processor to Run First Time

1. Select **Configuration** -> **Processor**
2. Locate the Setup Processor
3. Hit the **play** button: 
4. The execution count will increase and the total execution time will increment

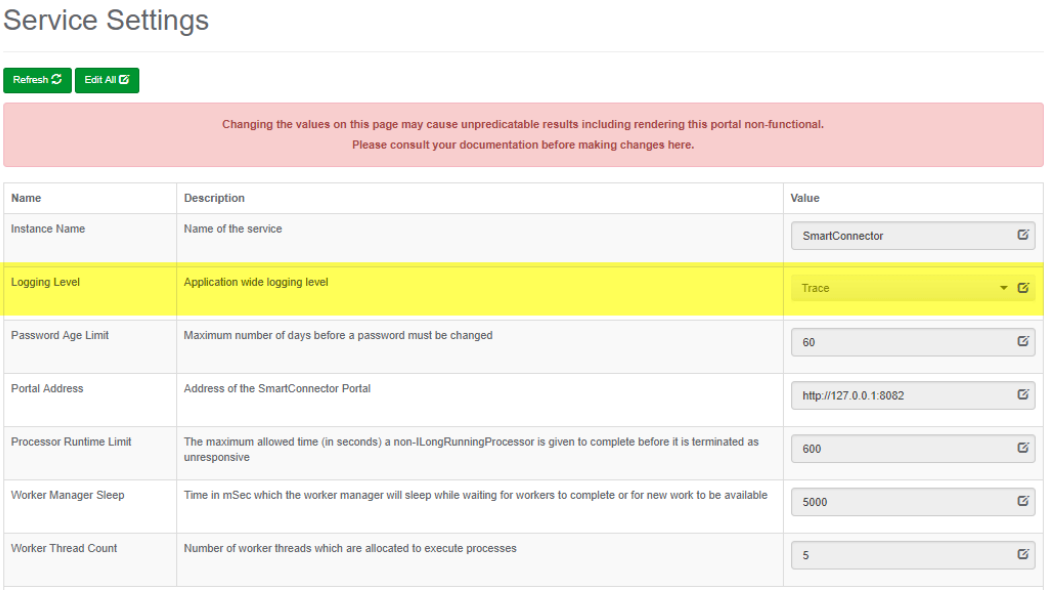
4. Troubleshooting

4.1. Smart Connector Log File

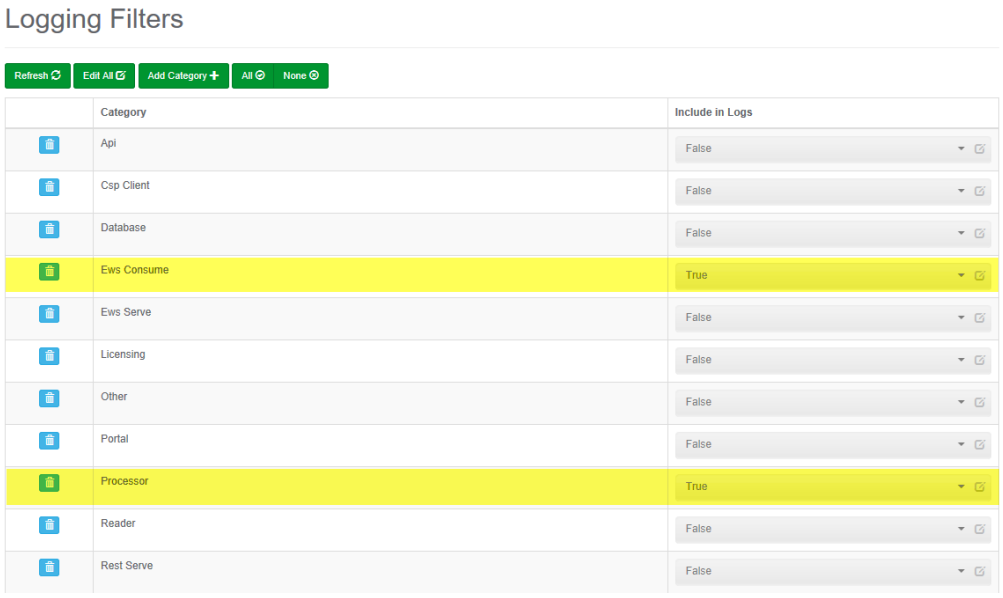
Smart Connector includes integrated logging into log files where both Smart Connector extensions and the Smart Connector framework can log any messages that may be useful. These log files can be found generally in the directory ***C:\\ProgramData\SmartConector\Logs*** on the machine where Smart Connector is installed.

In general, if you are having problems with Smart Connector or the EBO IoT Edge Smart Connector extension, it may be necessary to increase the logging level, or enable additional logging filters.

- To adjust the logging level, visit the **Service Settings** page and edit the *Logging Level setting.*

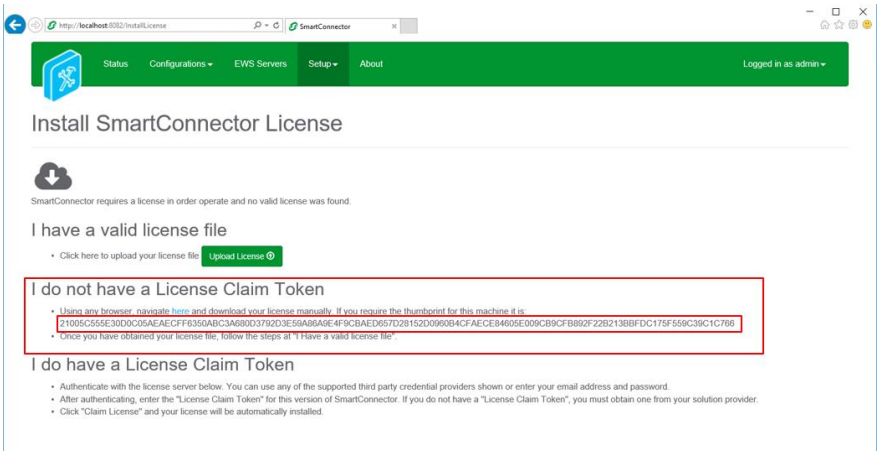


- To adjust the logging filters, visit the **Logging Filters** page. The logging filters most likely to pertain to this solution is ***Processor*** and ***Ews Consume***.



4.2. Framework Licensing Error

If you navigate to the Smart Connector portal and see a page similar to the below screenshot. This means that either you have not yet got a license for your Smart Connector framework, or your current license is no longer valid.



If you have not yet got a license for your Smart Connector framework:

Follow the instructions in the section Installing the Smart Connector Framework Runtime License

If you have already got a license for your Smart Connector framework:

The Smart Connector framework license is bound to a machine thumbprint. This machine thumbprint is a key generated from multiple hardware components of your machine, including the current network adapter that was being used when the license was generated. If you have switched to a different network adapter (e.g. going from a hard-wired connect to a WIFI connection), then it is very likely this machine thumbprint has changed. Please follow the section Installing the Smart Connector Framework Runtime License using your new thumbprint.

4.3. Smart Connector Extension Licensing Error

If your EBO IoT Edge Smart Connector extension processors are not running, please verify that they contain a valid license by:

1. Navigate to the processors configuration page.
2. Click on the 'Validate' button
3. If the error displayed is "License not found." You will need to obtain a license for the extension.

If you have not yet got a license for your EBO IoT Edge Smart Connector extension

Follow the instructions in the section Licensing the EBO IoT Edge Smart Connector Extension

If you have already got a license for your EBO IoT Edge Smart Connector extension

The EBO IoT Edge Smart Connector extension license is bound to a machine thumbprint. This machine thumbprint is a key generated from multiple hardware components of your machine, including the current network adapter that was being used when the license was generated. If you have switched to a different network adapter (e.g. going from a hard-wired connection to a WIFI connection), then it is very

likely this machine thumbprint has changed. Please follow the section Licensing the EBO IoT Edge Smart Connector Extension using your new thumbprint.

4.4. SQL Authentication Errors

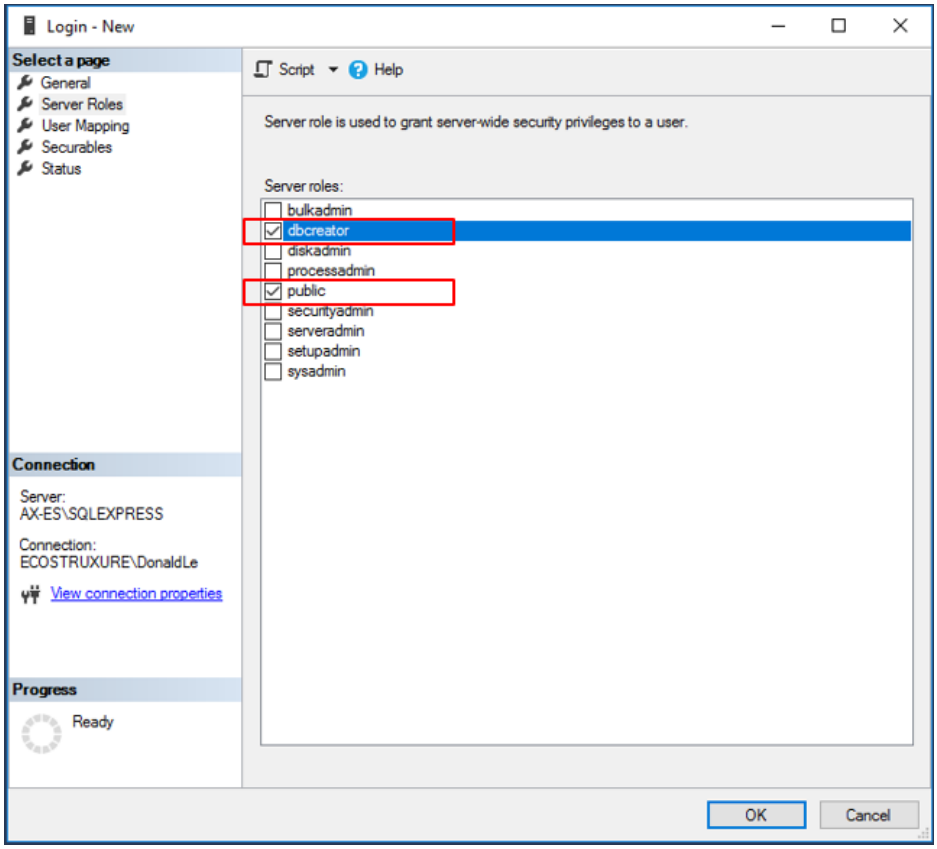
If Smart Connector cannot connect to its database, then the framework will fail to start. If you notice that the Smart Connector Server is not starting, or starting and instantly stopping, please review the Smart Connector logs for messages pertaining to SQL Authentication. If this is the case, you may need to make sure that your SQL Credentials are valid before starting the Smart Connector service.

4.5. MQTT Broker Communication Errors

If the EBO IoT Edge Smart Connector extension is unable to make a valid connection to the MQTT Broker, the Smart Connector log will display that this has occurred. If you are having problems where it seems you may not be getting the data that you expect, or no data at all. Please check the Smart Connector log for information about what may be going on.

5. Appendix A - SQL User Roles Definition

The Windows user installing the Smart Connector Framework software must have 'dbcreator' and 'public' roles within SQL in order for Smart Connector Framework to install correctly. During the installation process of Smart Connector Framework the database tables' necessary for configuring the system will be created.



Note: If the logged in Windows User did not have the proper SQL user roles during the installation process, the DB tables will not be created. You will need to *uninstall* then *reinstall* Smart Connector Framework to create the tables, once the Windows User has proper SQL roles defined. An attempt to perform an installation selecting "Modify" or "Repair" will not create the default DB for Smart Connector Framework.