Zebra RFID – EtherNet/IP

Sample Application

User Guide

MN-003923-03EN Rev. A

Terms of Use

Proprietary Statement

This manual contains proprietary information of Zebra Technologies Corporation and its subsidiaries (“Zebra Technologies”). It is intended solely for the information and use of parties operating and maintaining the equipment described herein. Such proprietary information may not be used, reproduced, or disclosed to any other parties for any other purpose without the express, written permission of Zebra Technologies.

Product Improvements

Continuous improvement of products is a policy of Zebra Technologies. All specifications and designs are subject to change without notice.

Liability Disclaimer

Zebra Technologies takes steps to ensure that its published Engineering specifications and manuals are correct; however, errors do occur. Zebra Technologies reserves the right to correct any such errors and disclaims liability resulting therefrom.

Limitation of Liability

In no event shall Zebra Technologies or anyone else involved in the creation, production, or delivery of the accompanying product (including hardware and software) be liable for any damages whatsoever (including, without limitation, consequential damages including loss of business profits, business interruption, or loss of business information) arising out of the use of, the results of use of, or inability to use such product, even if Zebra Technologies has been advised of the possibility of such damages. Some jurisdictions do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Contents

List of Figures

List of Tables

About This Guide

Introduction

This guide describes how to use the EtherNet/IP (EIP) sample application with the FX Series RFID reader using Studio 5000 Logix Designer.

IMPORTANT: If you have a problem with your equipment, contact Zebra Global Customer Support for your region. Contact information is available at: www.zebra.com/support.

Chapter Descriptions

Topics covered in this guide are as follows:

- Getting Started provides the pre-requisites to run the EtherNet/IP application with the RFID reader.
- Setting Up Sample App with Studio 5000 Logix Designer and PLC Controller explains how to setup Studio 5000 Logix Designer with the sample application and how to load the sample application to the PLC controller.
- RFID Configuration and Operations Using the EtherNet/IP (EIP) Sample App explains how to configure the reader with RFID specific configuration to perform Inventory and Access operations.
- Error Codes and Troubleshooting lists the error codes specific to the EtherNet/IP protocol and RFID operation, and includes information to troubleshoot common issues when running the application.

Related Documents and Software

The following documents provide more information.

- FX Series RFID Fixed Reader Integration Guide
- FX Series RFID Fixed Reader FX Connect Licensing Management User Guide
- RFID Reader Software Interface Control Guide
- EtherNet/IP Deliverable

For the latest version of this guide and all guides, go to www.zebra.com/support.

Notational Conventions

The following conventions are used in this document:

- **Bold** text is used to highlight the following:
  - Dialog box, window and screen names
  - Drop-down list and list box names
  - Check box and radio button names
  - Icons on a screen
  - Key names on a keypad
  - Button names on a screen.
- Bullets (•) indicate:
  - Action items
  - Lists of alternatives
  - Lists of required steps that are not necessarily sequential.
- Sequential lists (such as those that describe step-by-step procedures) appear as numbered lists.

Getting Started

Introduction

This chapter explains the pre-requisites and procedures to install the Studio 5000 Logix Designer; add the Zebra RFID reader AOP (Add-On Profile) extension to Studio 5000 Logix Designer; and, configure/use the sample application with custom RFID configurations.

Requirements

This section describes software and hardware requirements and how to use the Zebra FX9600 RFID reader with an EtherNet/IP industrial protocol.

Hardware Requirements

- Zebra FX9600 RFID Reader
- PLC compliant with EtherNet/IP

NOTE: The EtherNet/IP stack in the FX9600 RFID reader requires two I/O connections with the PLC and consumes approximately 6KB of memory for all data.

Software Requirements

- FX9600 RFID Reader updated with 3.6.21 build or later
- Zebra EtherNet/IP license installed on Reader
- Studio 5000 Logix Designer v32.0 or later
- AOP installed with Studio 5000 Logix Designer
- RSLinx Classic v32 or later
- Sample project downloaded from Zebra Support Central
- Project Rungs and Data Types from Zebra Support Central

Setting Up the FX9600 Reader with EtherNet/IP Application

The FX9600 reader is enabled with the EtherNet/IP stack with the embedded application. The EtherNet/IP application is available with FX9600 build version 3.6.21, and later, and can be downloaded as an installable Debian package. See Update the Reader Software on page 13 for information about updating the FX9600 reader with appropriate version.

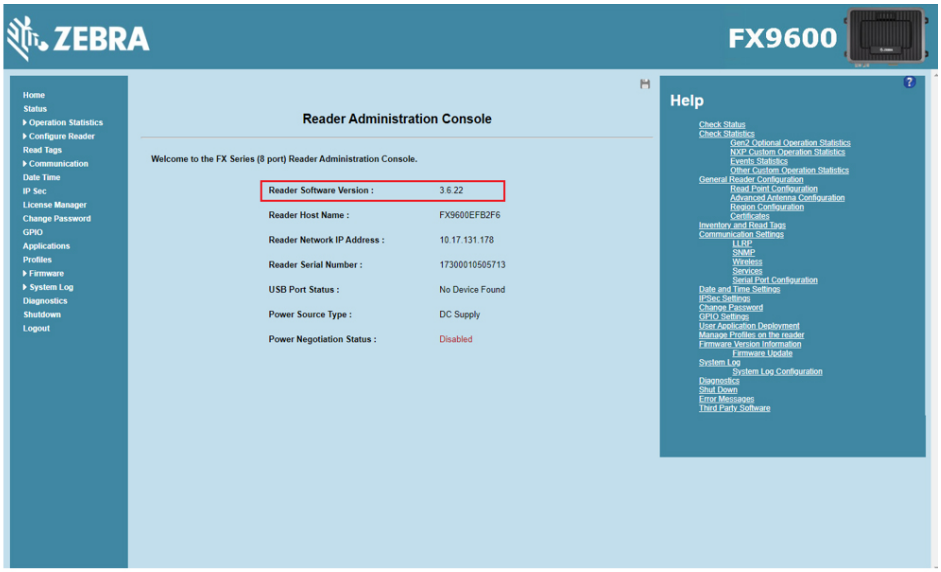
EtherNet/IP is a licensed feature and requires the installation of a Zebra EIP license before using the application with the Zebra FX9600 RFID reader. See Install EtherNet/IP License on page 13 for information about obtaining and installing the license.

When the license is successfully installed, the user can run the EIP application through the web console. See Install and Run the EtherNet/IP Application on page 14 for information about installing and running the application.

Update the Reader Software

To use the EIP functionality, the FX9600 Reader must run firmware version 3.6.21, or later. Verify the current running version from the FX9600 Reader web-console shown in Figure 1.

Figure 1 Reader Administration Console



If the reader is running an older software version, upgrade the reader with new version. Download the FX Series RFID Fixed Reader Integration Guide and follow the instructions in the section Firmware Upgrade.

Download the latest software for the FX9600 Reader at:

zebra.com/us/en/support-downloads/software/operating-system/fx9600-series-operating-system.html.

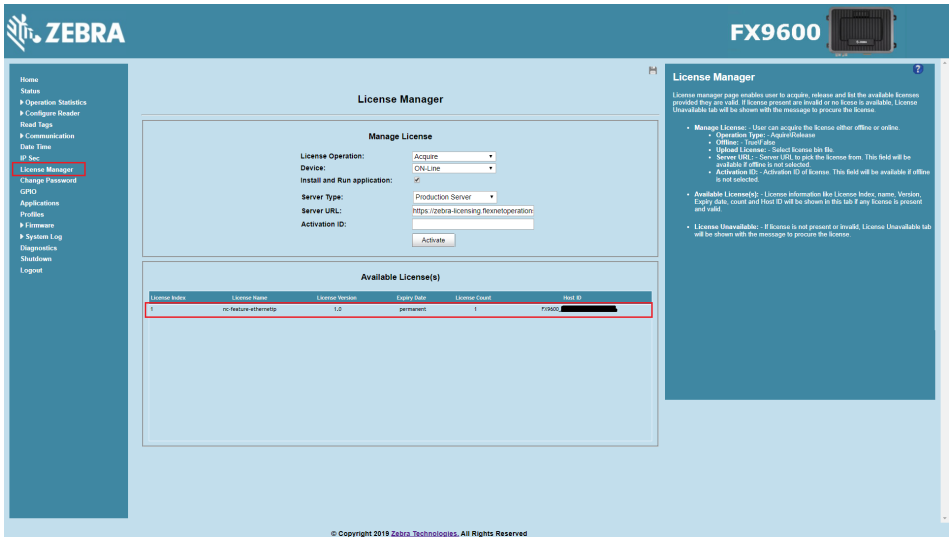
When the reader runs the appropriate software version, install the EIP license to run the application.

Install EtherNet/IP License

Install the EIP license (for EtherNet/IP) from the FX9600 Reader web-console. Follow the instructions in the FX Series RFID Fixed Reader FX Connect Licensing Management User Guide to request, obtain, and install the license for Zebra FX Series readers.

When the license is installed, the licensing information can be viewed from the web-console shown in Figure 2.

Figure 2 License Manager



Install and Run the EtherNet/IP Application

The EIP application can be installed one of two ways:

- along with the license by enabling the check-box Install and Run application

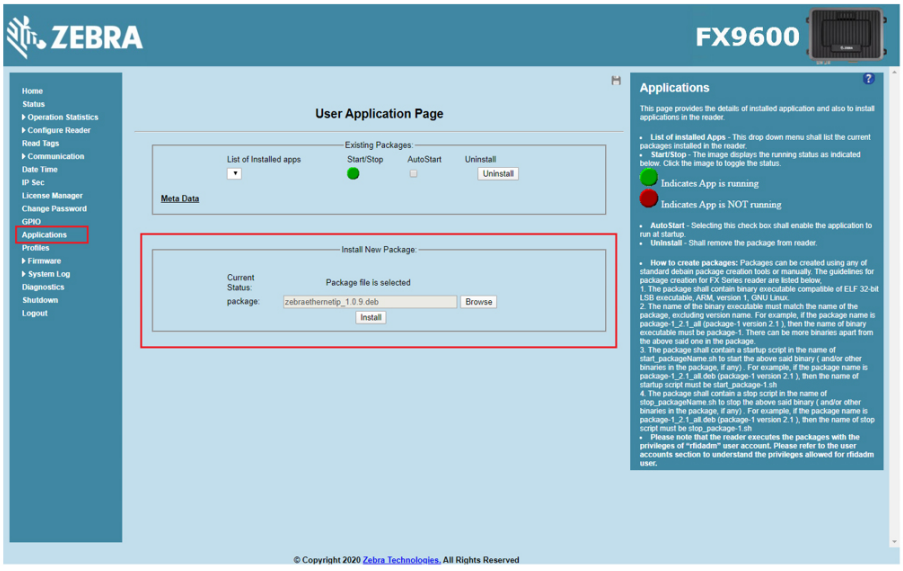
  or

- by installing the application with the web-console as a Debian package.

To install the application as a Debian package:

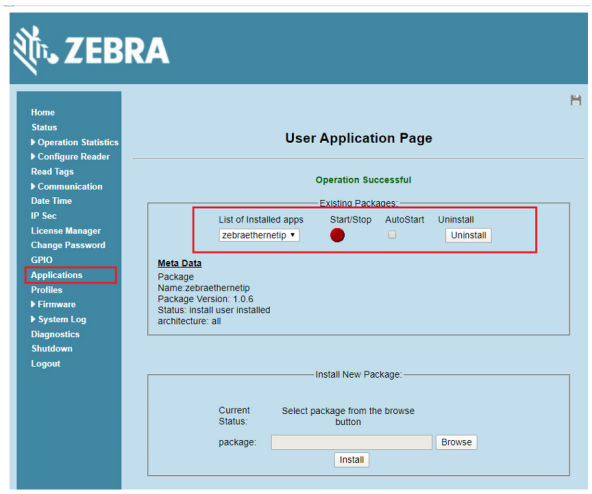
1. Navigate to the **User Application Page** and browse for downloaded application Debian package.

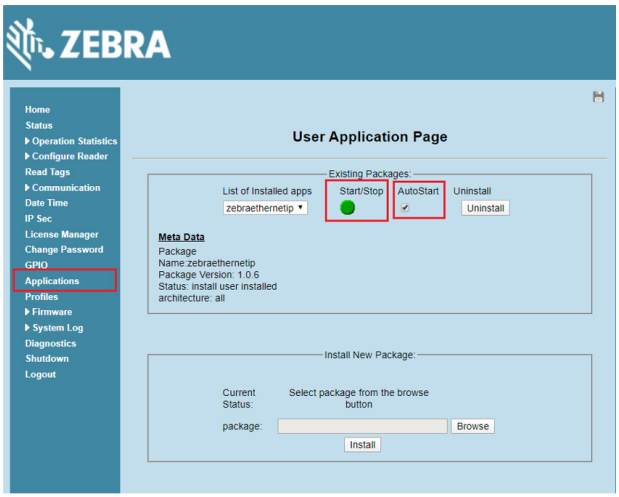
Figure 3 User Application Page



2. Click **Install**. When the application is installed, it is available under Existing Packages in the List of Installed apps drop-down list.

3. To run the application, click the red **START/STOP** button The button turns green to show the application status as running. The user can also choose to AutoStart the application on reader reboot by selecting the AutoStart check-box.





NOTE: The web-console session logs out upon application start-up. The user must login to the web-console again to access the reader.

4. When the application starts, PLC Controller connects with it and is ready to perform RFID operations through the EtherNet/IP protocol.

Setting Up Sample App with Studio 5000 Logix Designer and PLC Controller

Installing Studio 5000 Logix Designer

If not already installed, the user must install Studio 5000 Logix Designer to use the sample application. Studio 5000 Logix Designer provides a means to load the program into PLC and to communicate with PLC to perform operations using sample application. Follow the installation instructions from Rockwell Automation. A license may be required for Studio 5000.

Download the studio 5000 Logix Designer version from Rockwell Studio at:

rockwellautomation.com/global/products/factorytalk/overview.page?pagetitle=Studio-5000-Logix-Designer&docid=924d2f2060bf9d409286937296a18142.

Installing Zebra RFID Add-On Profiles Extension

There are custom Add-On Profiles (AOP) extensions which can be used with Studio 5000 Logix Designer. Install the Zebra RFID AOP as a Studio 5000 extension.

An AOP is used within Studio 5000 Logix Designer to define a specific piece of hardware and how it reacts within the control system. AOP is the customized extension and it provides the capability to configure the project for specific hardware with predefined control parameters.

Once AOP is installed, the Zebra RFID module will be available in Studio 5000 Logix Designer. The designer creates Zebra RFID specific EtherNet/IP assembly objects for input and output assemblies and creates supporting data structures.

Download the Zebra RFID AOP extension at:

zebra.com/us/en/support-downloads/rfid/rfid-readers/fx9600.html.

Once downloaded, double click the installer and follow on-screen instructions to add the Zebra RFID AOP extension to Studio 5000 Logix Designer. System may need to restart after installing AOP.

Installing Zebra RFID Module using EDS File

For the PLC that does not support AOP, use the EDS file to configure the Zebra RFID EtherNet/IP on the PLC. The EDS file comes with the FX9600 Industrial Ethernet Software package, which is available at:

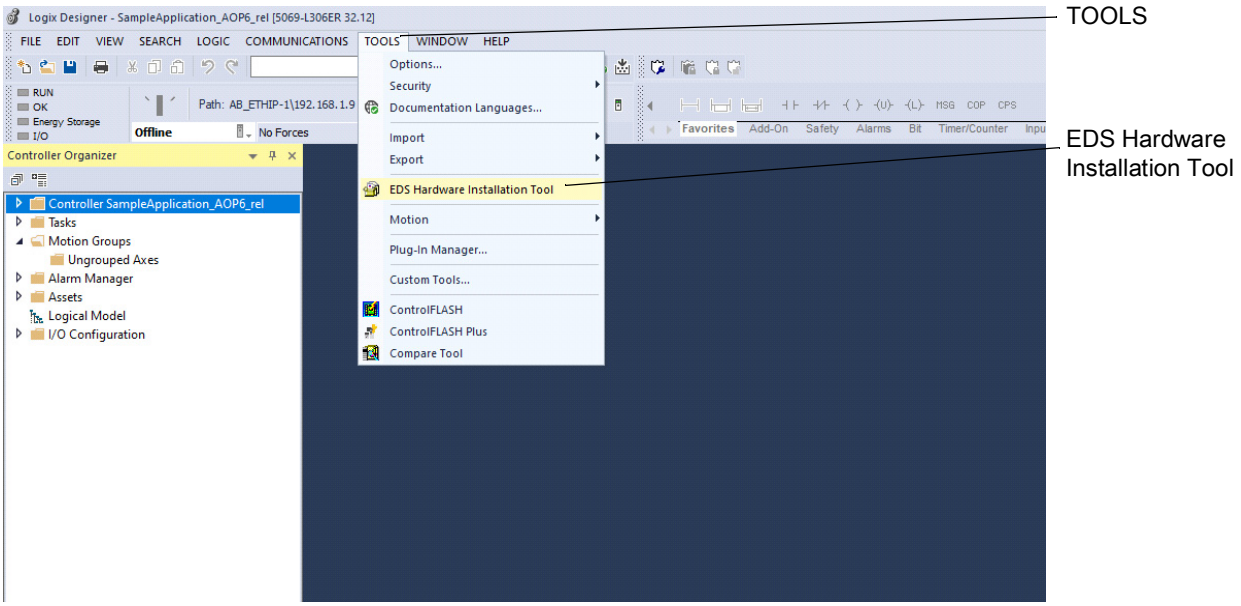
zebra.com/us/en/support-downloads/rfid/rfid-readers/fx9600.html

Once the EDS file is extracted, you can load it using Studio 5000 Logix Designer or any supported tool for the EtherNet/IP.

To install the EDS file using Studio 5000 Logix Designer:

1. Select **EDS Hardware Installation Tool** from the **TOOLS** drop-down menu.

Figure 4 EDS Hardware Installation Tool Option



2. Follow the steps in the subsequent wizard dialog boxes.

Using RFID-EIP Sample Application

Once Studio 5000 Logix Designer and custom Zebra RFID AOP are installed the user can open the sample application with Studio 5000 Logix Designer. The sample application is created using Studio 5000 Logix Designer v32.12 and CompactLogix 5380 Controller (Model 5069-L306ER).

IMPORTANT: There are some limitations with Studio 5000 Logix Designer that might require modifications in the sample application. If Studio 5000 Logix Designer and the PLC versions are not compatible with the setup, see Using Non-compatible Versions of Studio 5000 Logix Designer and PLC on page 17 for details.

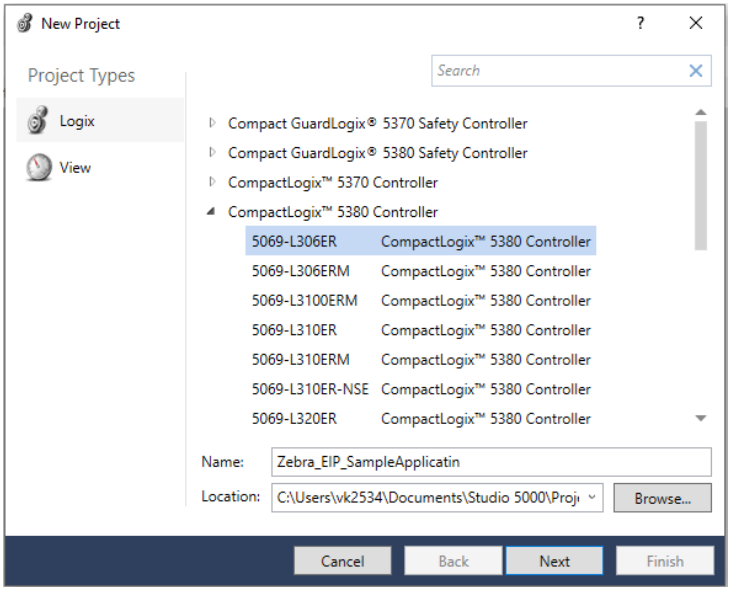
With compatible versions of Studio 5000 Logix Designer and the PLC, the user can open the sample application with a (dot)ACD extension in Studio 5000 Logix Designer. See Configuring the Application on page 25 for application configuration.

Using Non-compatible Versions of Studio 5000 Logix Designer and PLC

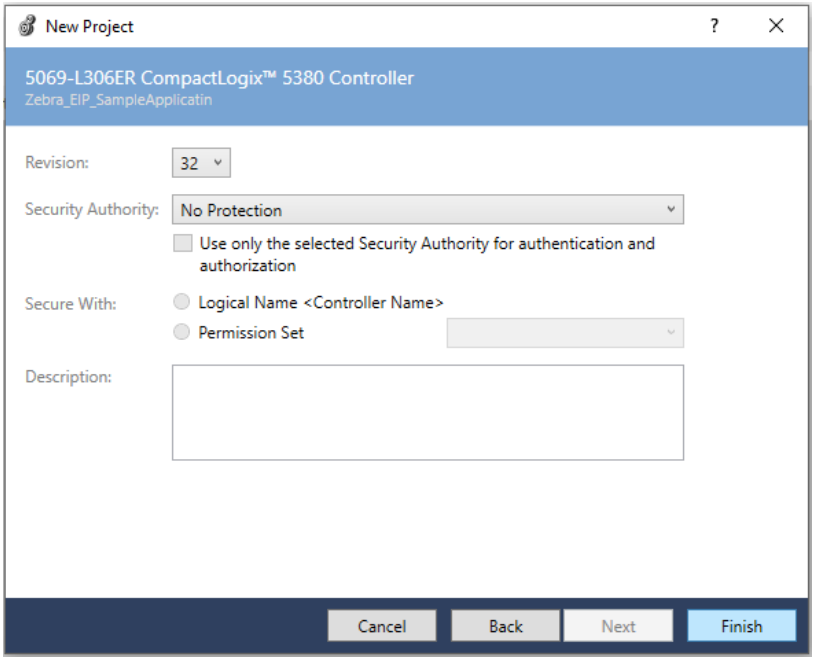
If Studio 5000 Logix Designer and the PLC versions are not compatible, the user needs to create a new project to import the Rungs which are available in the sample application package. Follow the steps below to create a new project and import the Rungs.

1. Start Studio 5000 Logix Designer and choose to create a new project.

2. Choose the appropriate controller and provide name of the project as shown below.

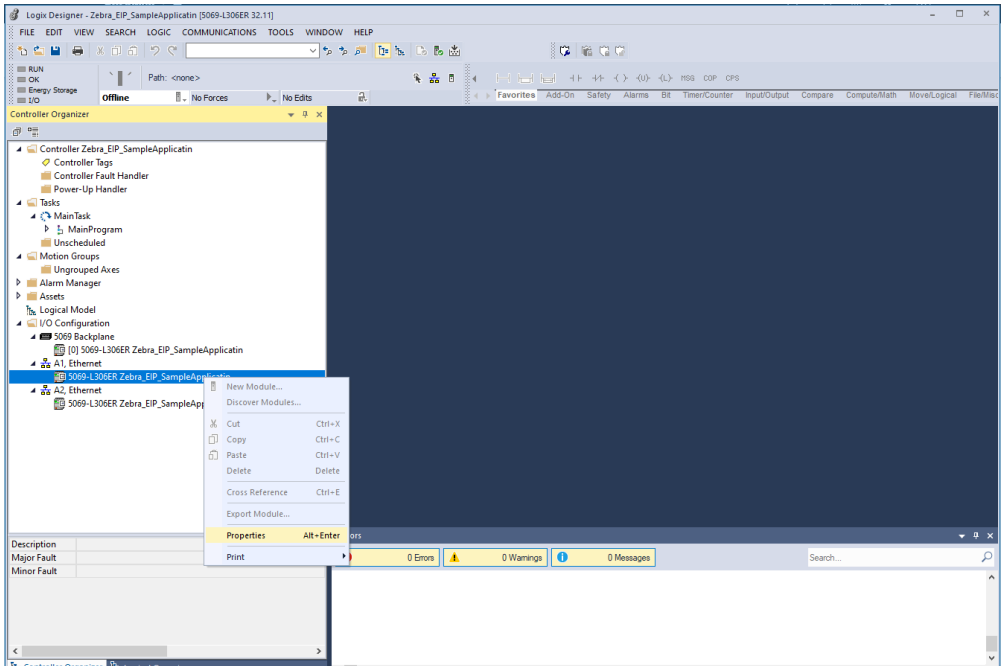


3. Click **Next**. The **New Project** dialog box displays.



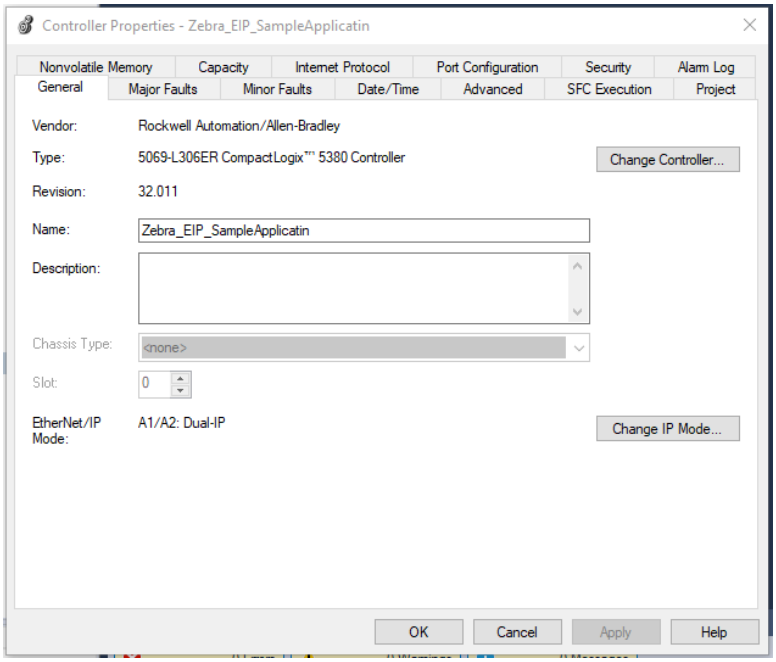
4. Leave the default options as is and click **Finish**. A new empty project generates.
5. Configure the controller properties by right clicking on **Ethernet Connection** from **I/O Configuration** in left panel.

Figure 5 New Empty Project



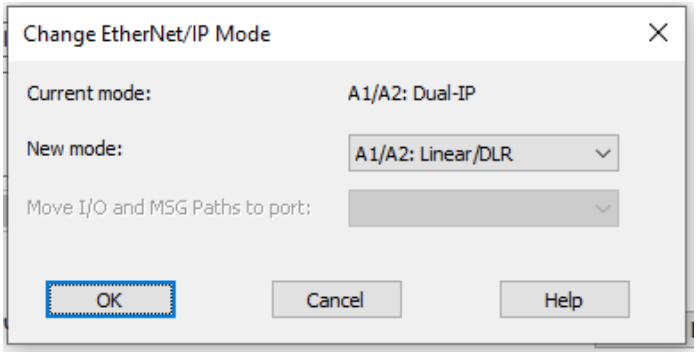
6. The user can edit the controller configuration as per their controller and requirement.

Figure 6 Controller Properties



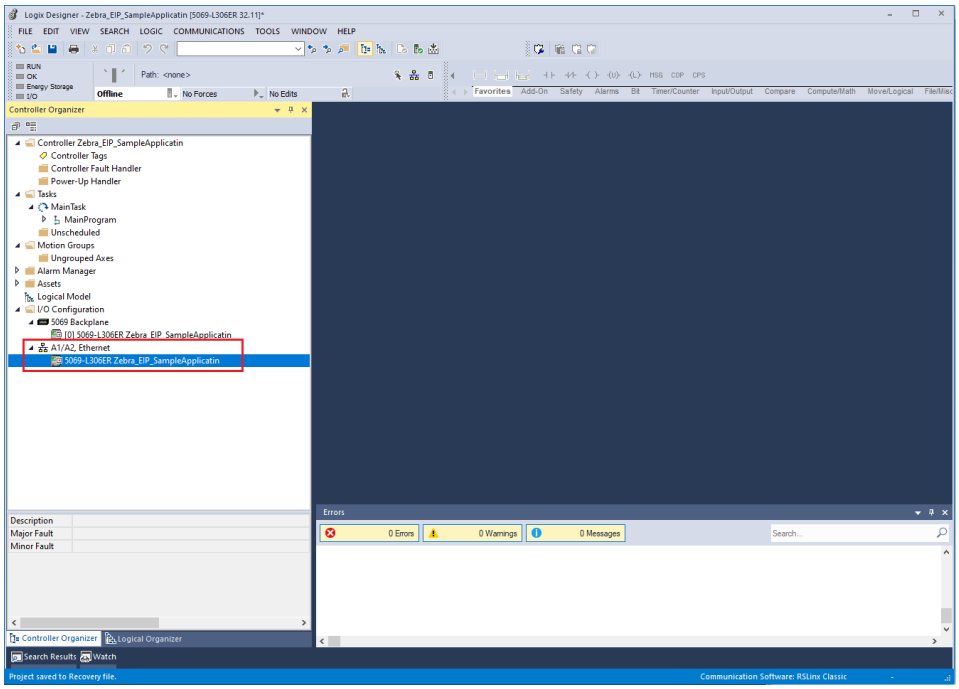
7. The sample application uses Linear IP mode for EtherNet/IP. Click **Change IP Mode..** . The **Change EtherNet/IP Mode** dialog box displays.

Figure 7 Change Ethernet/IP Mode



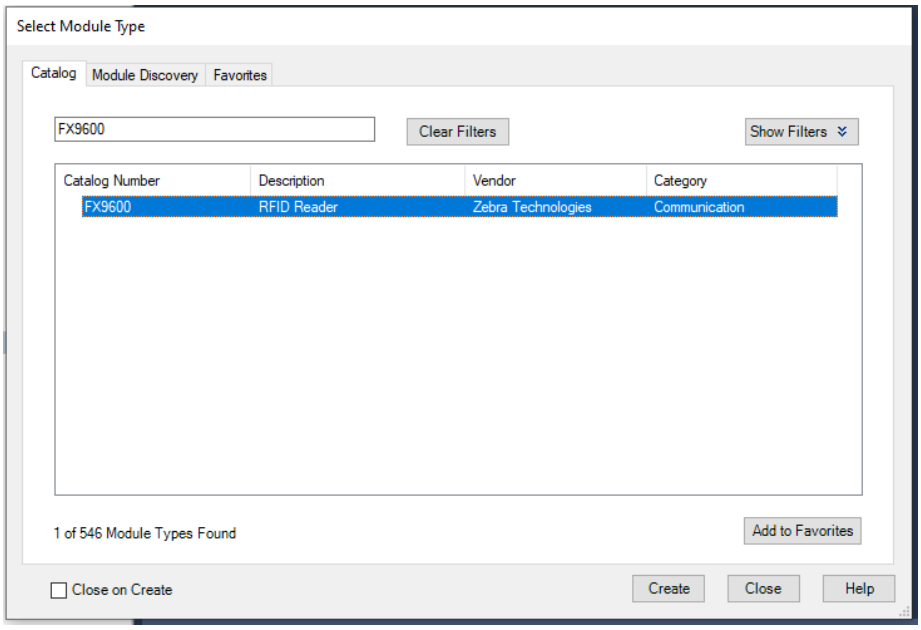
8. In the **New mode:** drop-down box, change the IP mode from Dual IP to linear.
9. Click **OK** in the Change EtherNet/IP mode dialogue.
10. Click **OK** ion Controller Properties dialogue.
11. The controller mode is changed to Linear IP mode.

Figure 8 Linear IP Mode

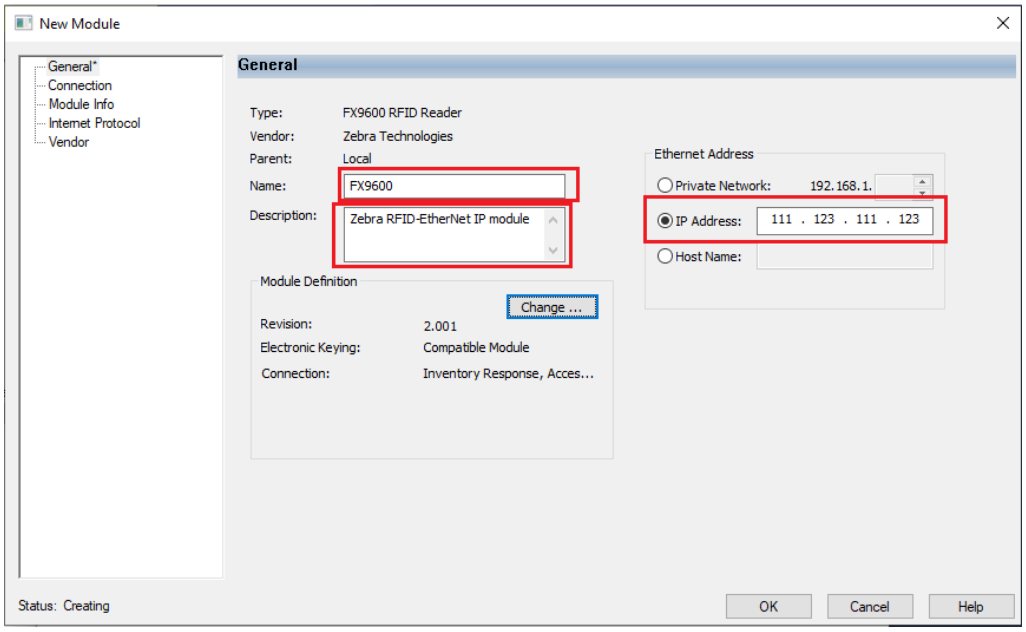


12. Add the new RFID module to the project for RFID assembly operation by right clicking **Ethernet Controller** under **I/O Configuration** in the left plane.
13. Select New Module to open a module selection dialogue with all available modules.
14. In the filter text box, enter **`FX9600`** to display the Zebra specific RFID Reader Module (FX9600).

Figure 9 Module Type



15. Select the FX9600 module and click **Create**. The **New Module** dialog box displays.



16. Add the module related information.

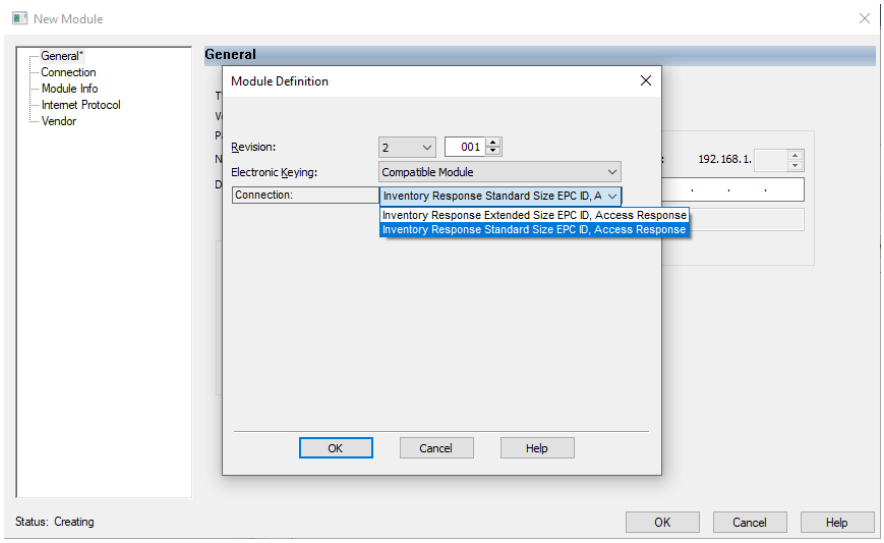
a. **Name**: Name of the module as per your choice

b. **Description**: Module Description (optional)

c. **IP Address**: IP Address of RFID Reader

d. The user can also choose the module definition (two available configurations) by clicking **Change...** below **Module Definition**. The **Module Definition** dialogue window displays and the user can choose the connection from the drop-down menu.

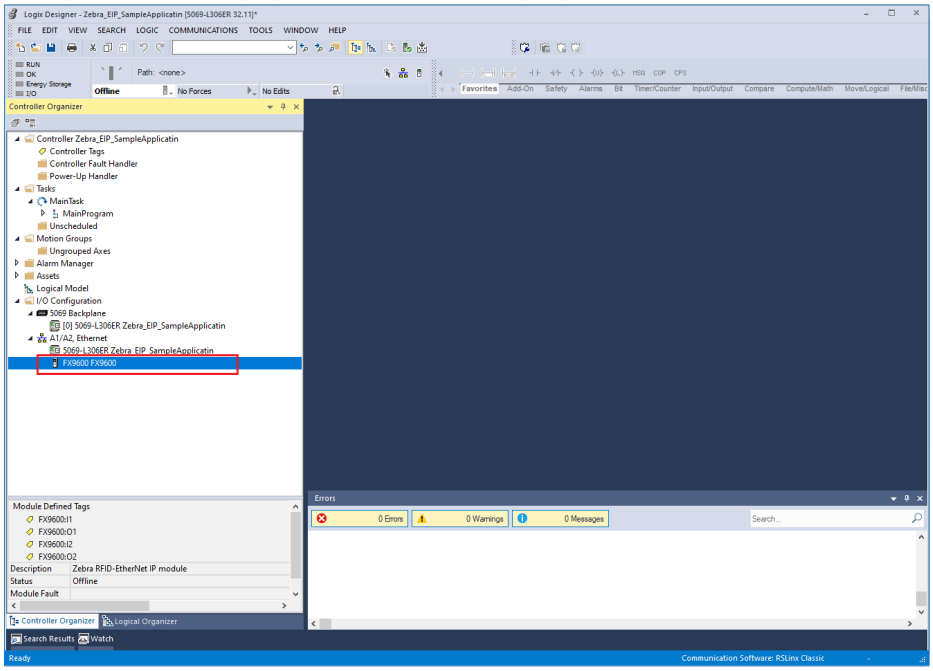
Figure 10 Module Definition



Select Inventory Response Extended Size EPC ID, Access Response or Inventory Response Standard Size EPC ID, Access Response. Both connections support Inventory and Access operations. The main difference is in the inventory response. The first connection can be used when the user needs to support TAGs with max 64 Bytes EPC length. The second connection can read up to 12 Bytes EPC length only, which is the standard EPC ID length. The advantage of using standard EPC length is that more EPC IDs (more tags) can be reported per each refresh cycle.

17. Click **OK**. (In this example, Inventory Response Standard Size EPC ID, Access Response was selected.)
18. Click **OK** again in the New Module dialogue window.
19. Click on **Close** to close the module selection dialogue. The new Zebra RFID module is added in the project.

Figure 11 Newly Added RFID Module

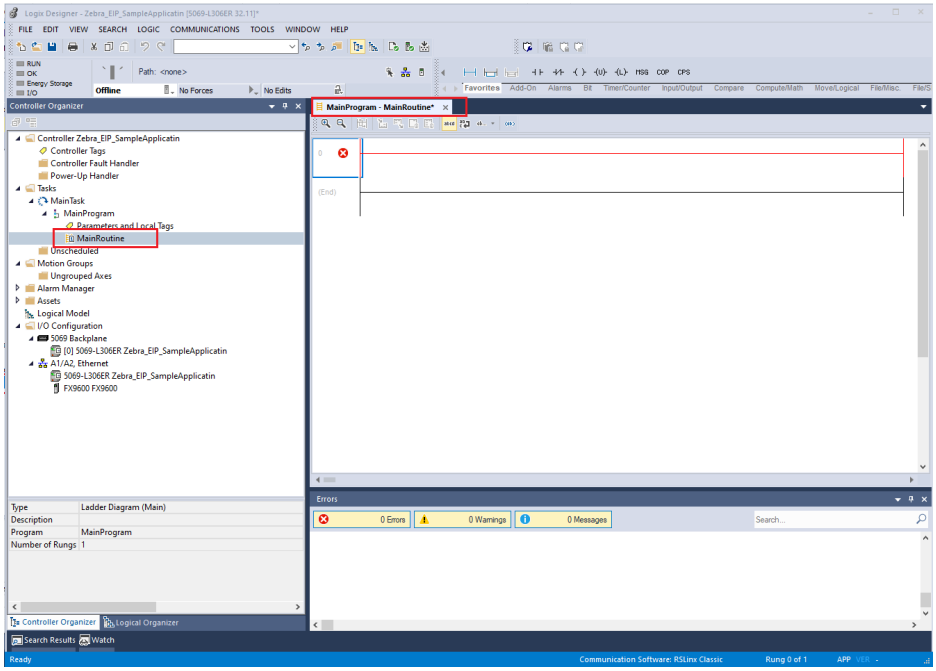


Importing Rungs

The module is now ready with an I/O connection object created in controller Tags. To complete the project, the user must import Rungs.

1. From the left panel, expand to **Tasks > MainTask > MainProgram**. This displays the **MainRoutine**. From the left panel, double click **MainRoutine** to display the **MainProgram - MainRoutine** in the right pane.

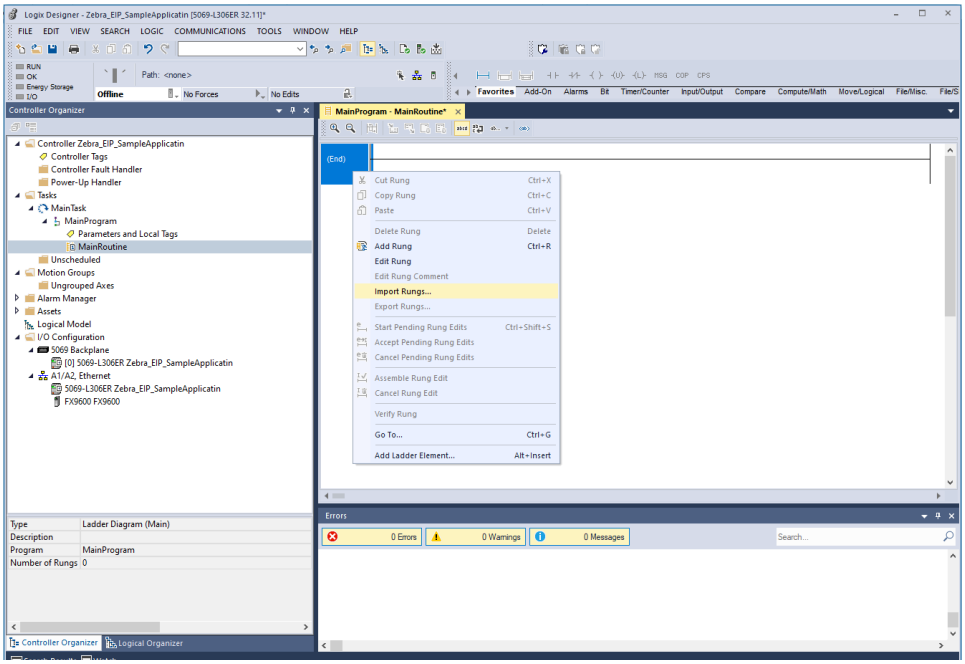
Figure 12 MainProgram - MainRoutine



2. By default, MainRoutine has no Rungs and they must be exported from the sample application.

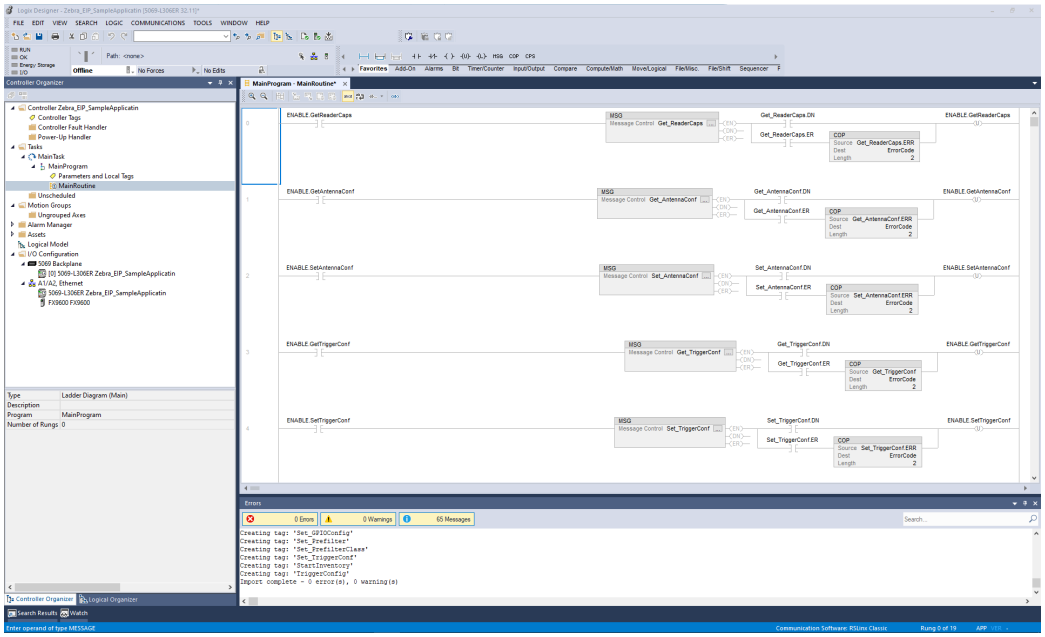
   Right-click on the default 0 Rung and select **Delete**.

Figure 13 Import Rungs



3. Right-click again on (End) and select **Import Rungs...** . The **Import Rungs** window displays.

Figure 14 Import Rungs Window



4. Select the exported Rung file with the L5X extension available with the sample app and click **Open**. Before the Rung import starts, click **OK** to confirm the import. The Rungs are imported and are available in the current program.
5. Save the program.
6. The sample application is available in Studio 5000 Logix Designer. Configure the application to work with current program.

Configuring the Application

The Studio 5000 Logix Designer project is directly loaded to the PLC Controller, so there is a need to specify an IP address of the reader and Controller Type so that PLC can communicate with reader. For this reason the sample app needs to be configured with the specific IP address in the RFID module.

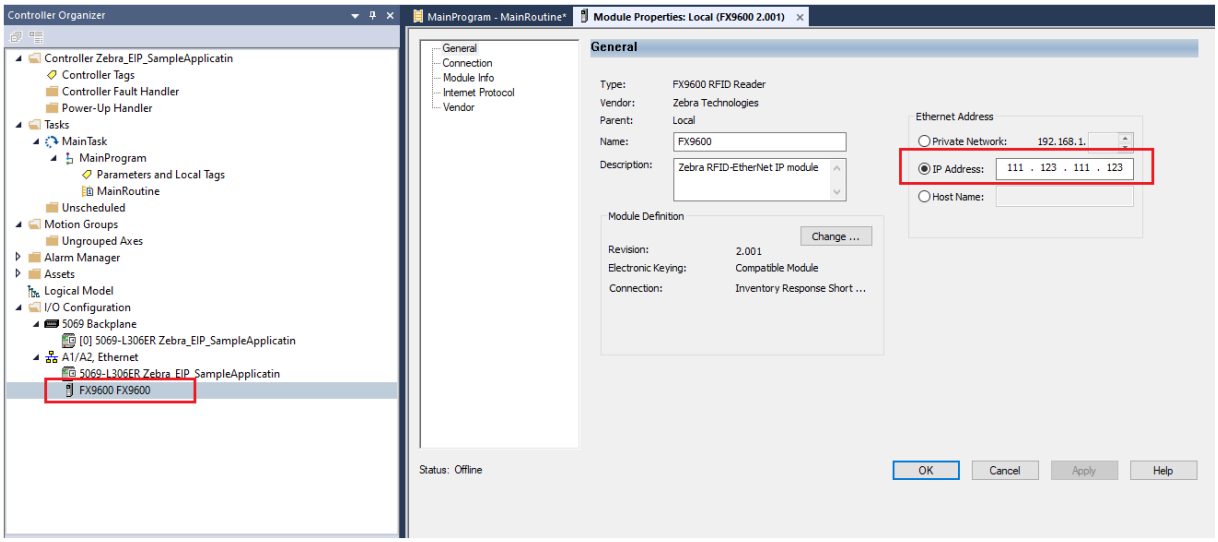
Users setting up the project from scratch do not need to do this as their project is already configured with the reader IP address. In this case, go to Updating the Configuration for MSG Instructions on page 26.

Setting the Reader IP With the Module

To configure the module to the reader IP, edit Module Properties and specify the correct IP address of the reader.

1. Right click on the FX9600 Module from left pane and select the Properties.
2. The module properties dialogue opens in the right pane. From that pane, select the General tab and modify the IP address to the correct reader IP address as shown below.

Figure 15 Module Properties



3. Click **OK**.

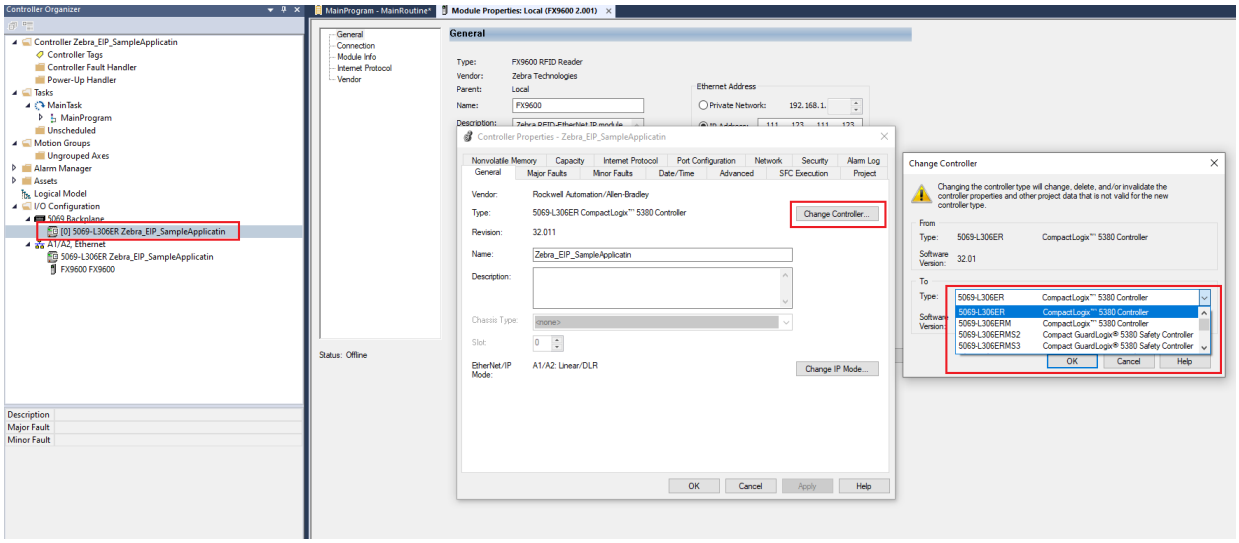
Configure For the Appropriate Controller

If the controller is not a CompactLogix 5380 – 5069-L306ER the user must change the controller in the sample application project.

To change the controller in sample application:

1. Right click on the controller under **I/O Configuration > 5069 Backplane**.
2. Select **Properties**. The **Controller Properties** dialog window displays.

Figure 16 Change Controller



3. Click **Change Controller** to display the Change Controller dialog box.
4. Select the appropriate controller from the **Type** drop-down list.
5. Click **OK**.

Updating the Configuration for MSG Instructions

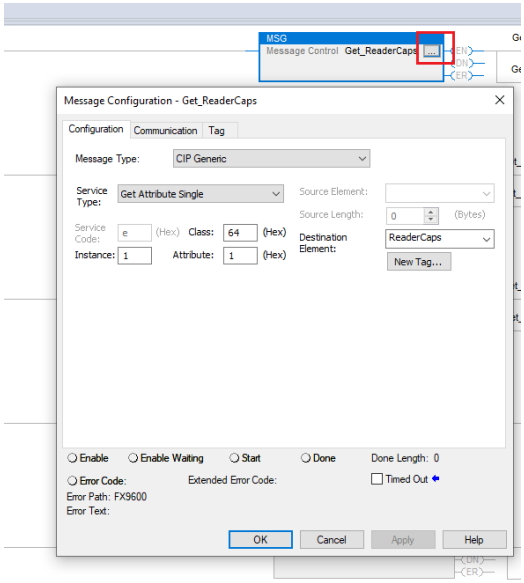
The RFID-EtherNet/IP model uses explicit messaging to apply the reader configuration. The sample application uses the Studio 5000 Logix Designer MSG instruction to perform explicit messaging. Each MSG instruction needs configuration to specify the module with which to communicate. But, as the module configuration is different from the sample application (Reader IP Address), each MSG instruction requires a configuration update.

NOTE: The following steps must be performed for each MSG instruction.

To update the MSG instructions with current module configuration:

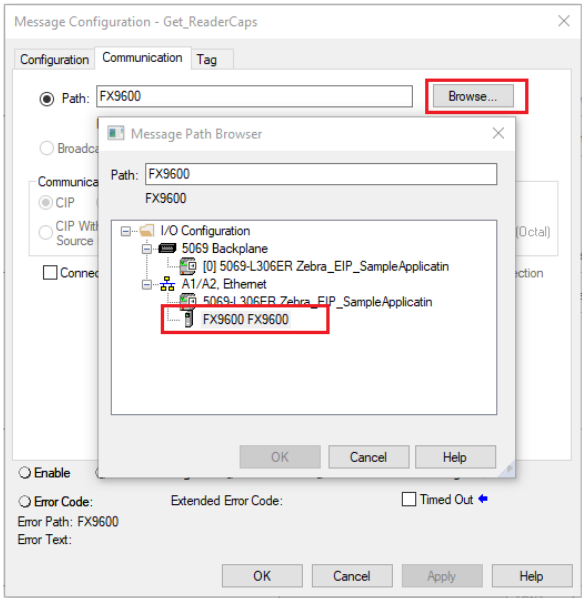
1. Select the MSG instruction and click the three dots in the upper right corner of the instruction (Configure button). The **MSG instruction Configuration** dialog box displays.

Figure 17 Configuration Dialog Box



2. Select the **Communication** tab.

Figure 18 Communication tab



3. Click **Browse** to display the **Message Path Browser**.
4. Select the FX9600 module as the message path and click **OK**.
5. Click **OK** again to save the message configuration.

Creating Application using Generic EtherNet/IP (without AOP)

This section describes the steps and parameters required to configure PLC controller for working with FX9600 RFID Reader using Generic EtherNet/IP module.

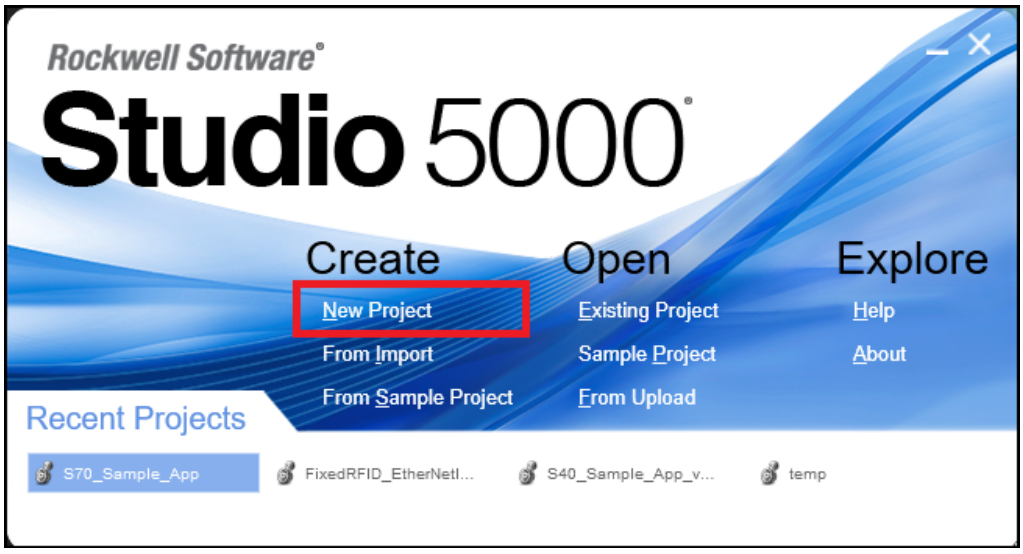
Below steps are demonstrated to create Generic EtherNet/IP module using Studio 5000 Logix Designer, but similar steps can be used to configure PLC controller on any other tools which supports EtherNet/IP module configuration.

Creating New Project with Correct PLC Controller

Create New Project by choosing correct PLC controller.

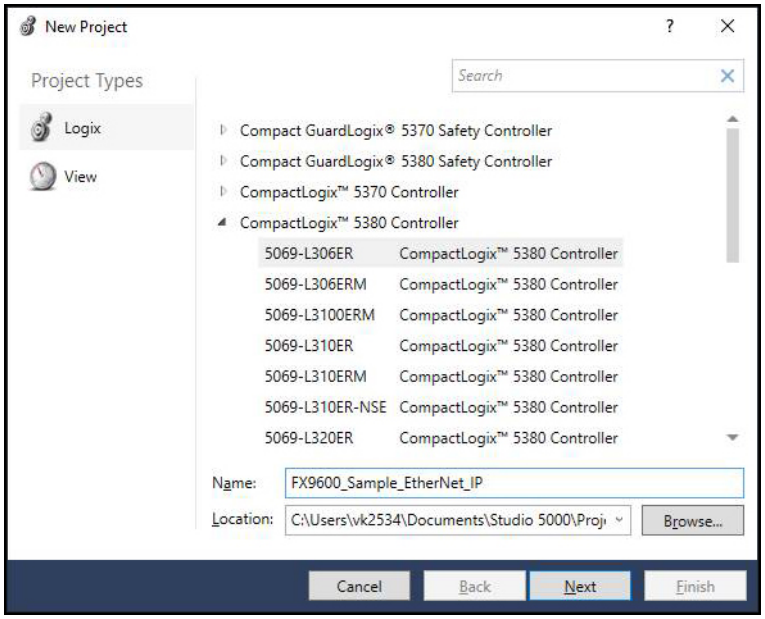
1. Open **Studio 5000 Logix Designer** application.

Figure 19 Studio 5000 Application



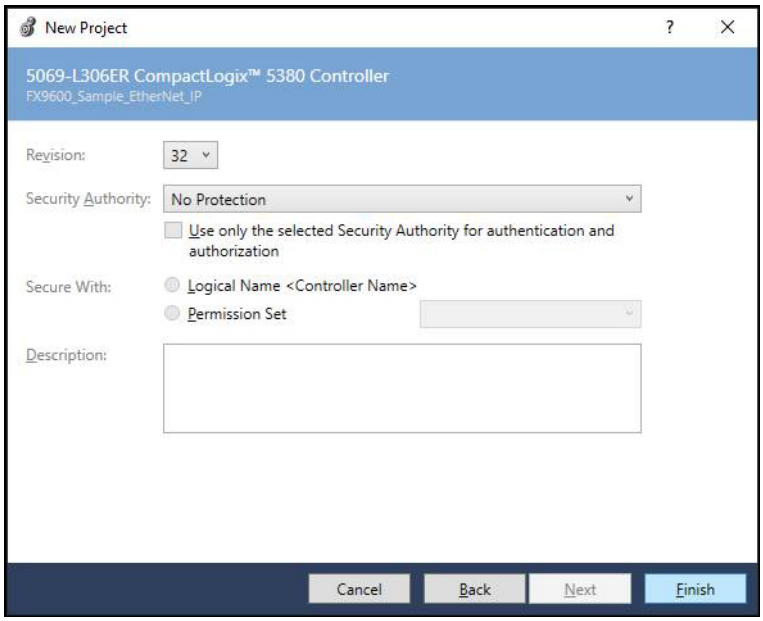
2. Select **New Project**. The **New Project** dialog box displays.

Figure 20 New Project Window

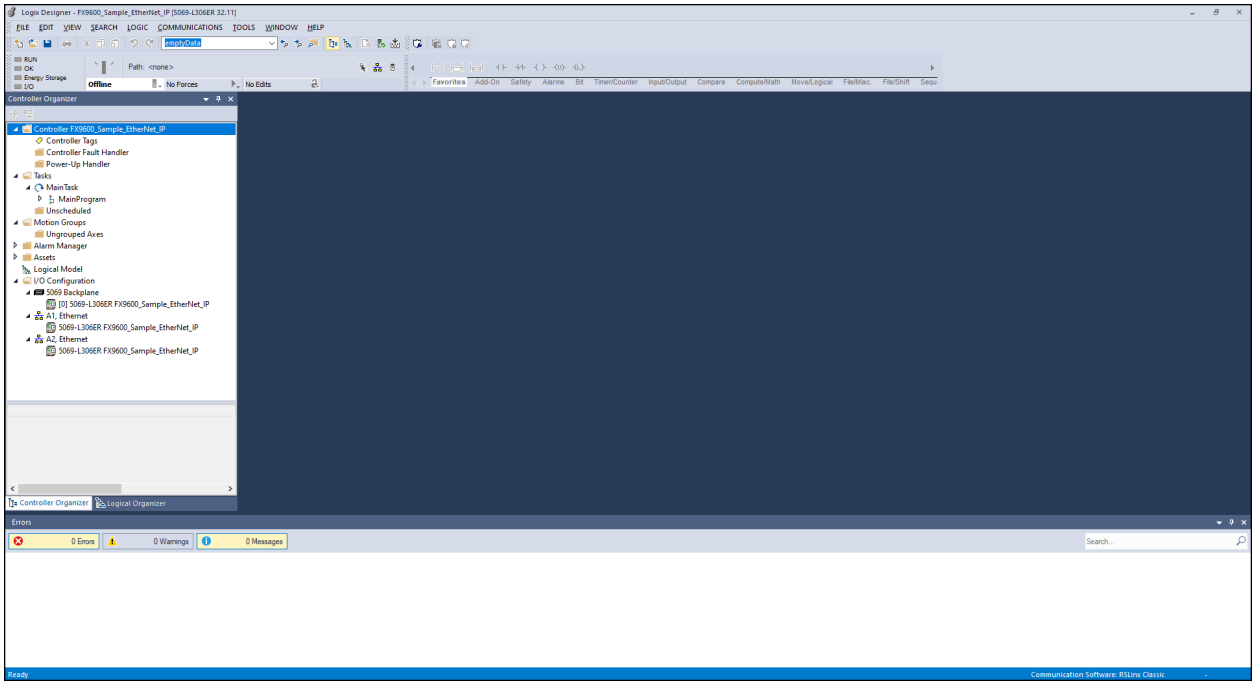


3. Choose the appropriate controller.
4. In the **Name** field, enter the project name.
5. Click **Next**.

Figure 21 New Project Defaults



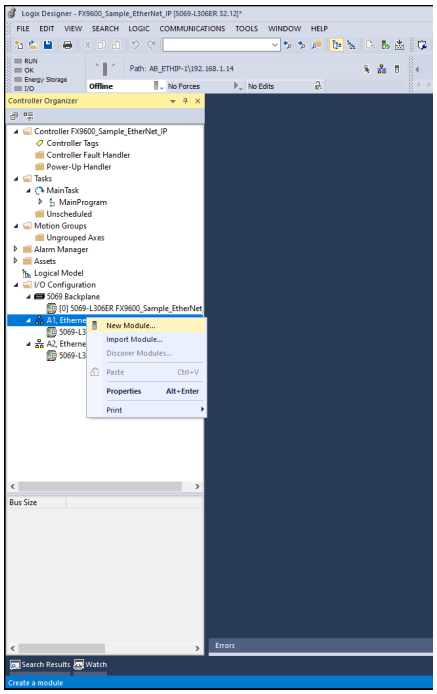
4.Leave the default options as it is and click **Finish**. It will create and open the new project.



Adding and Configuring EtherNet/IP Module

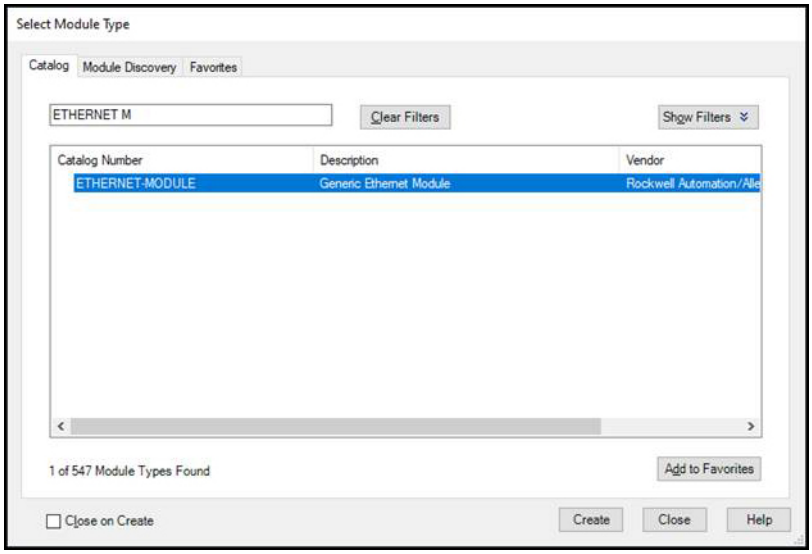
Add Generic Ethernet Module for Targeted device (for example: FX9600).

Figure 22 Create New Module



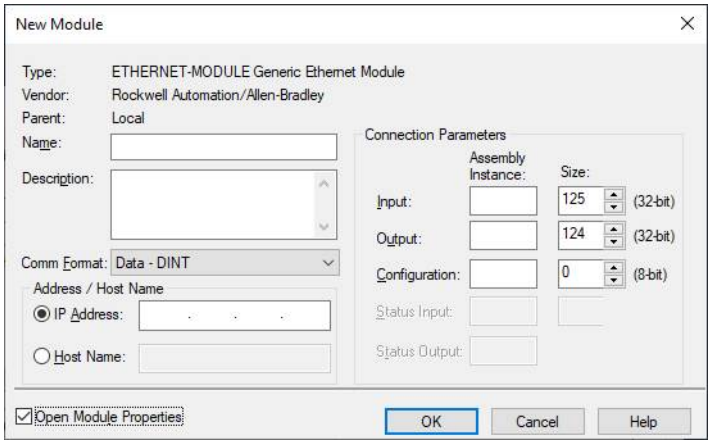
1. In the left panel, right click on the PLC controller and select **New Module**. The **Select Module Type** dialog box displays.

Figure 23 Select Module Type



2. In the search box, enter **`ETHERNET MODULE`**. ETHERNET MODULE appears in the window.
3. Select **ETHERNEMET MODULE**.

4. Click **Create** to create a generic module for EtherNet/IP. The **New Module** window displays.



5. Please follow next section to configure the module.

Configuring EtherNet/IP Module

NOTE: Assembly Instance ID and size is defined in Object Model document available at Zebra support site.

The FX9600 RFID Reader EtherNet/IP supports two pairs of I/O assemblies:

- Inventory operation - the reader can perform inventory operation and provide information of read TAGs.
- Access operation - the reader can perform read/write memory bank information of selected TAG.

The below example is created for inventory operation. The same steps can be followed to create user defined data structure for access operation assemblies as well following the Object Model. Inventory operations FX9600 RFID Reader EtherNet/IP as well supports two type of responses:

- Standard Length EPC TAG reporting: Standard length EPC TAG assembly supports up to 12 Byte EPC ID length of TAG. (Input Assembly Instance # 101)
- Extended Length EPC TAG reporting: Extended length EPC TAG assembly supports inventory operation on TAG with ECP length up to 64 Bytes. (Input Assembly Instance # 100)

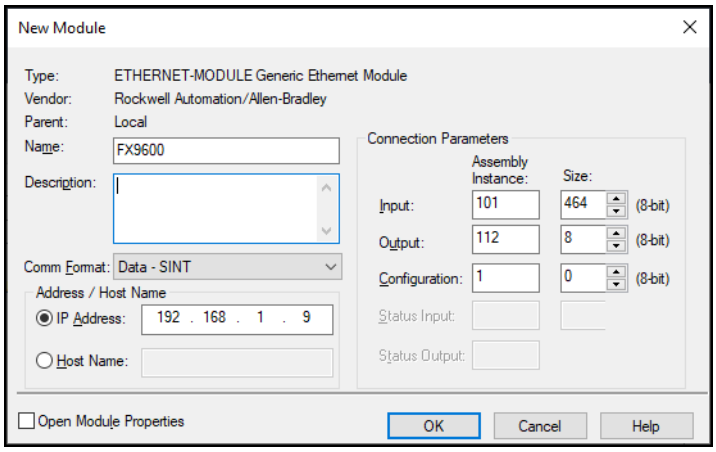
The user can choose any of the type by providing corresponding assembly id in Input Assembly Instance.

Configure I/O Assembly in Module in Module Creation Window by Providing Assembly Class IDs and Size

Example:

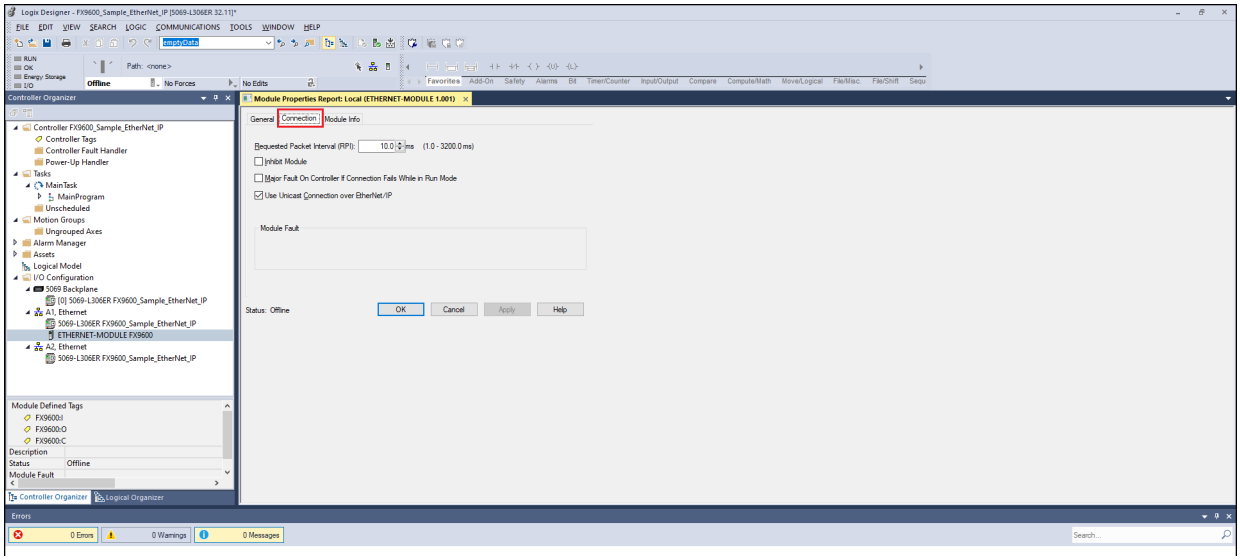
1. In the **Name** field, enter a name for the module.
2. In the **Input Assembly Instance** field, enter **101** for Standard Length EPC TAG.
3. In the **Input Assembly Size** field enter **464** bits.
4. In the **Output Assemble Instance** field, enter **112** for Inventory command.
5. In the **Output Assembly Instance** field, enter 8 bits.
6. In the **IP Address** field, enter the IP address of the target FX9600 RFID Reader.

Figure 24 New Module Configuration



7. In the **Comm Format** drop-down list, select **Data - SINT**.
8. Click **OK**. The module is added in the newly created project.
9. Close the **Module** selection window.
10. Once module is created, in the left pane, select the module properties.
11. Double-click on the module name.
12. Select the **Connection** tab.
13. In the **Requested Packet Interval (RPI)** field, select the RPI value per the requirement. The minimum RPI supported for FX9600 RFID Reader EtherNet/IP module is 10 ms.
14. Click **OK**.

Figure 25 RPI Selection



Create User Defined Data Structures

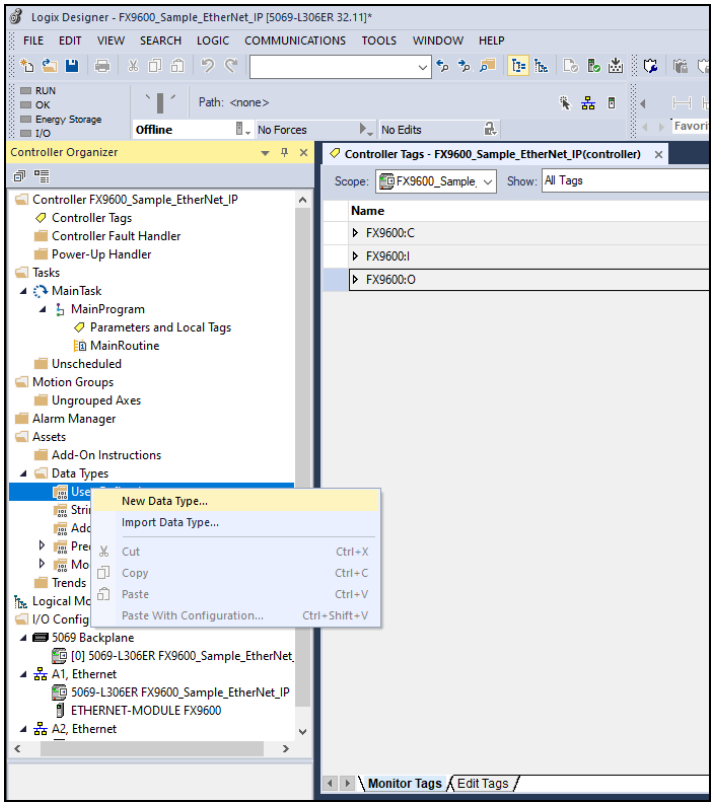
To read the data read from byte array read through PLC in meaningful manner it is required to typecast (copy the data in structured data variable). This can be done by creating custom data types and copy the data in those data types with proper byte boundaries.

These data structures are also defined in Object Model with corresponding data types as well.

To create a new data type:

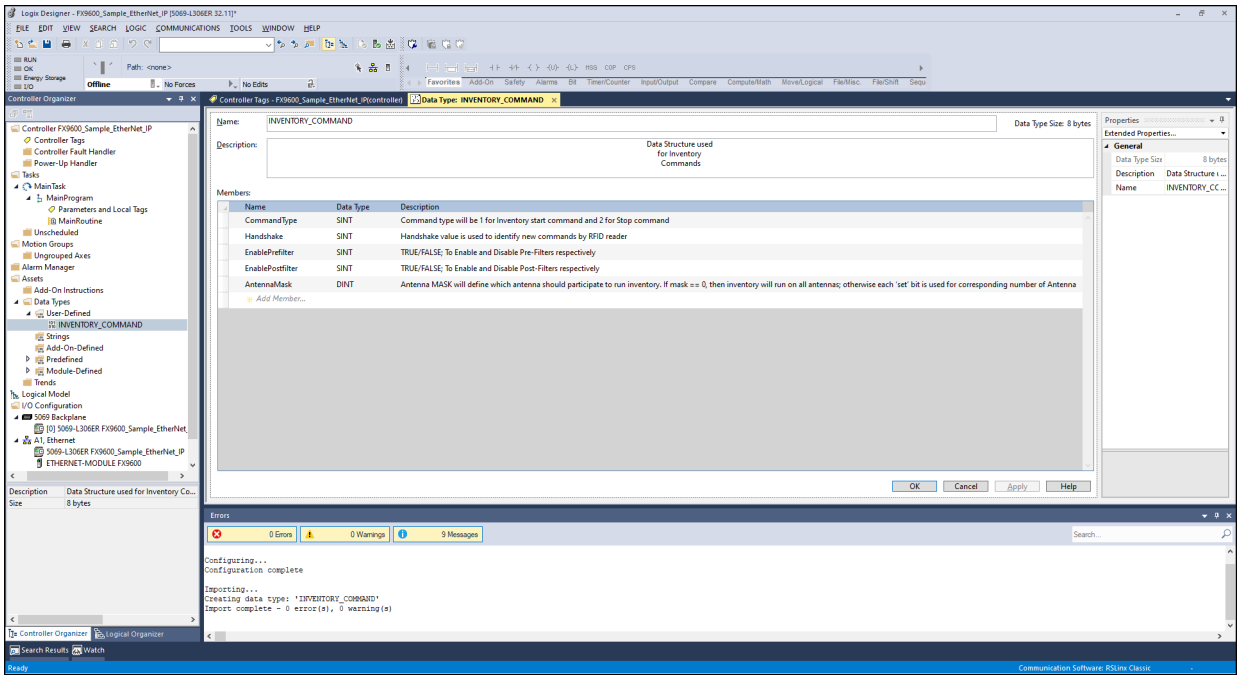
1. In the left pane, navigate to **Assets** > **Data types** > **User-Defined**.

Figure 26 New Data Type



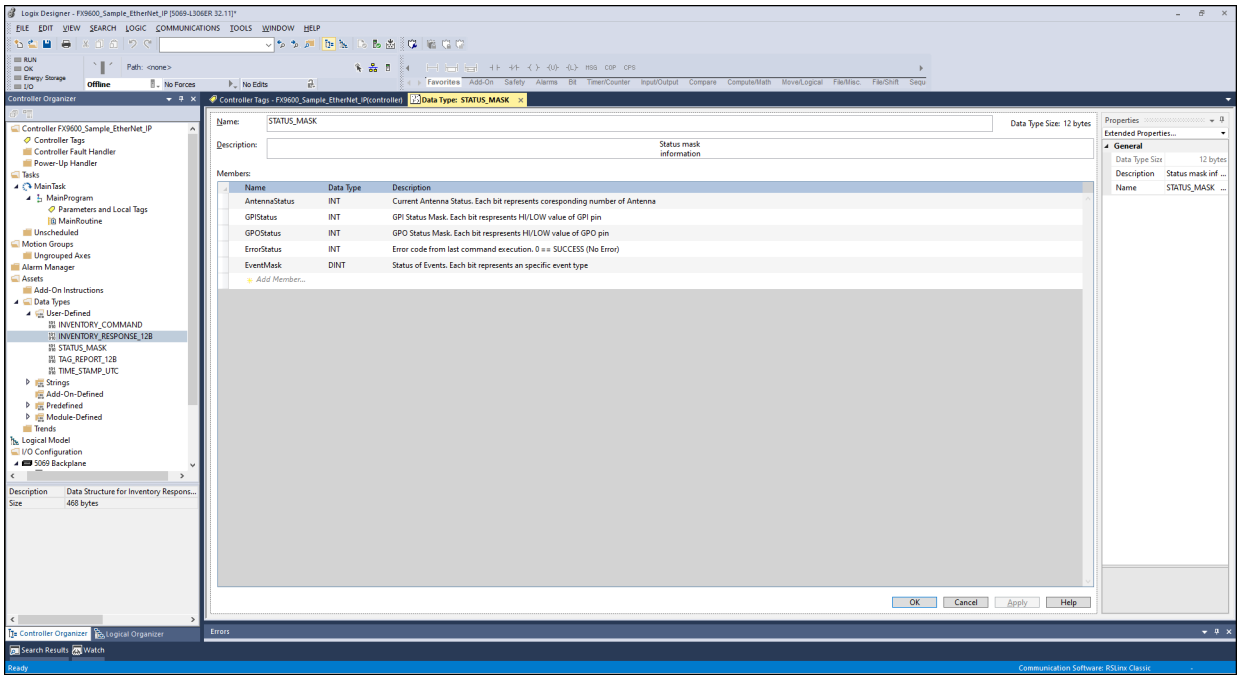
2. Right click on **User-Defined** and select **New Data Type**.
3. Create an Inventory Command Data Structure with the details in the below image.

Figure 27 Inventory Command Data Structure



4. Click **Apply**.
5. Click **OK**.
6. Follow the same steps as above and create Inventory Response Data Structure as shown in below images.

Figure 28 STATUS_MASK Data Structure



NOTE: Inventory response is created with multiple custom data structures. First, create those data structures as shown in below images. Each image describes a custom data structure with field description.

Figure 29 TIME_STAMP_UTC Data Structure

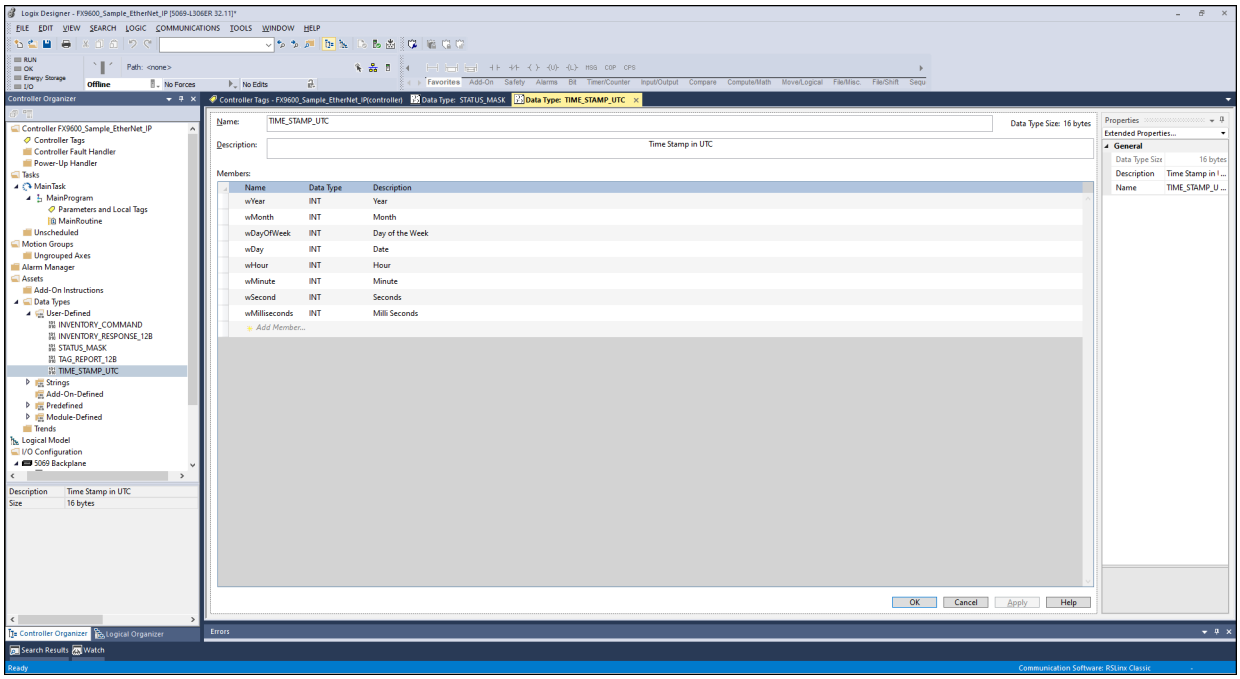


Figure 30 TAG_REPORT_12B Data Structure

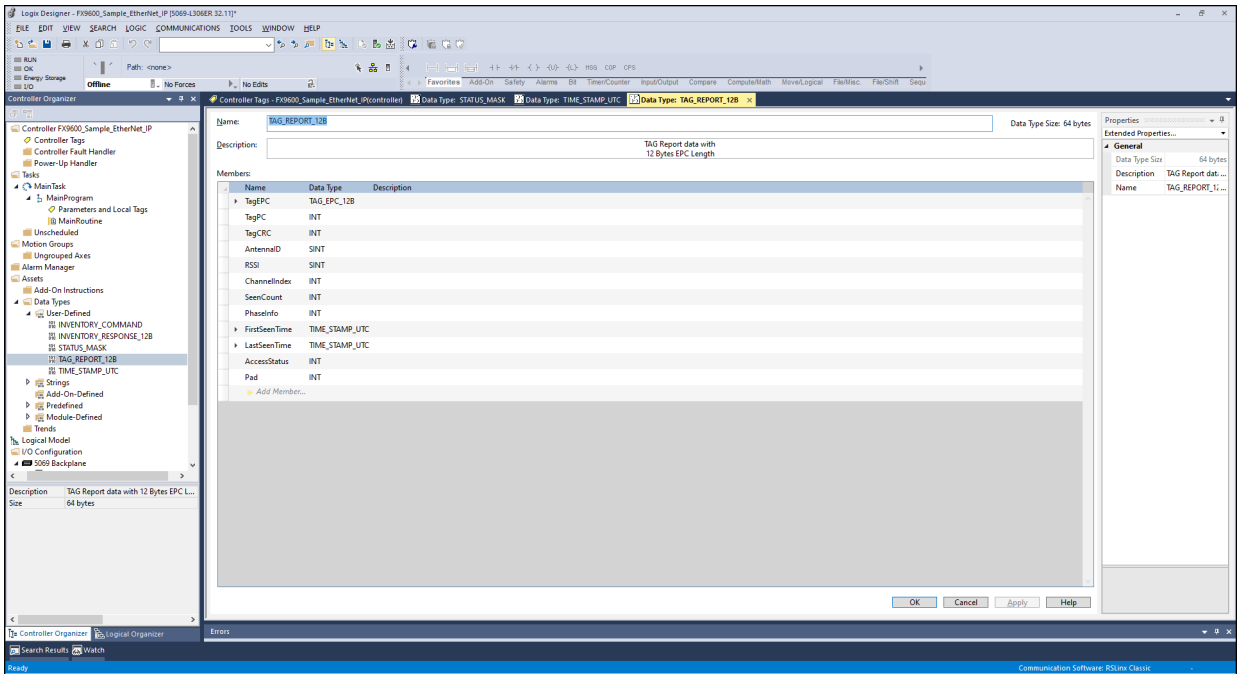
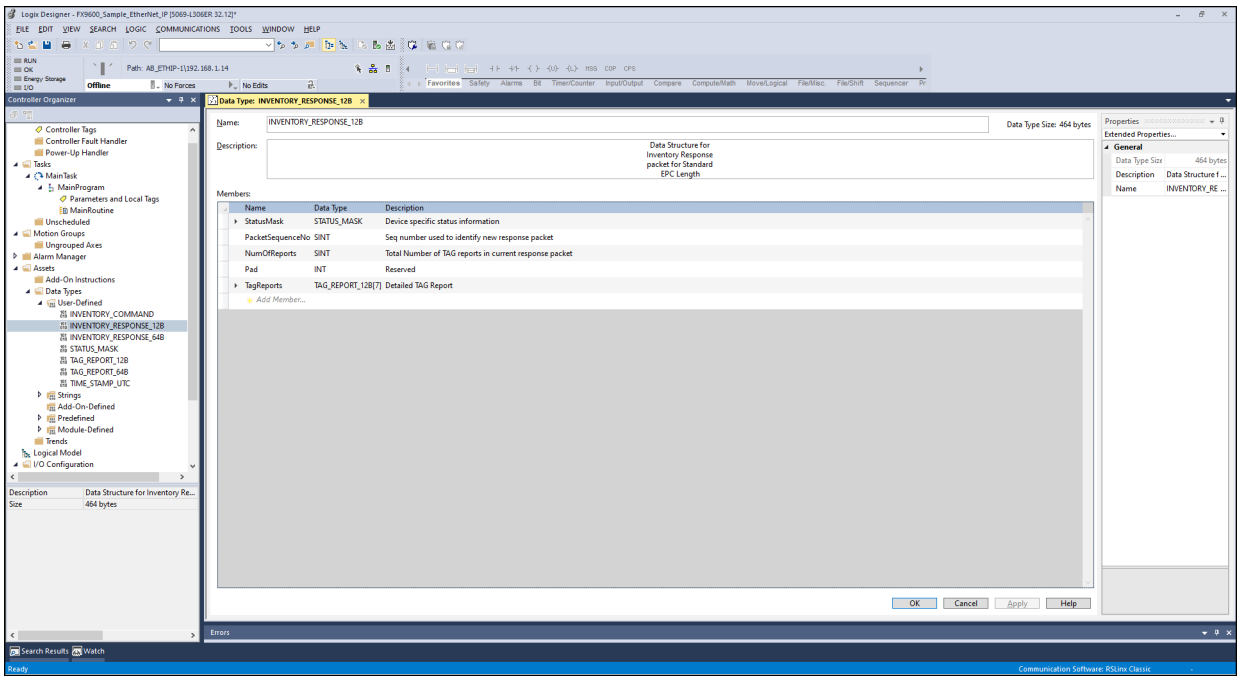


Figure 31 INVENTORY_RESPONSE_12B Data Structure



Sample Ladder Logic to Start/Stop Inventory Using I/O Assemblies

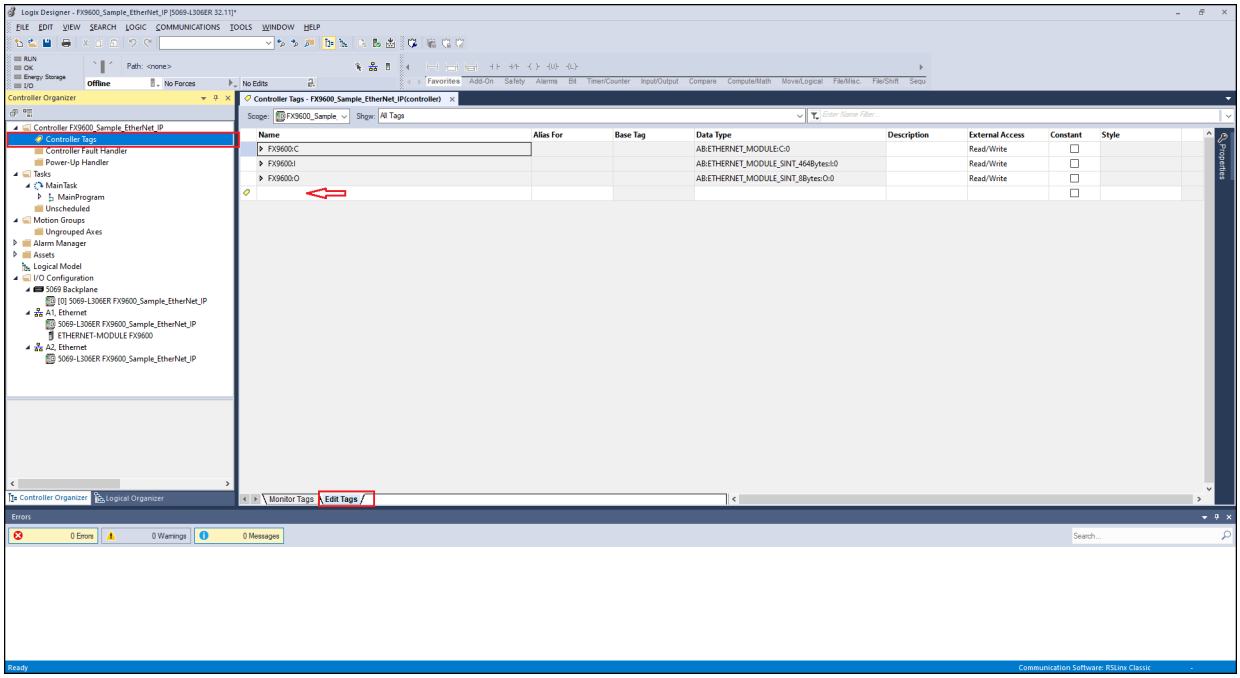
Create Required Controller Tags to control the operations in ladder logic as below:

- EnableOperation: This is of type BOOL and will be used to enable/disable specific RUNGs.
- InventoryCommand: This tag will be used to copy command data to Output Assembly instance. This is of custom data type "INVENTORY_COMMAND" created in previous section.

- InventoryResponse: To read the response from input assembly instance, it is required to copy data into a structured data type "INVENTORY_RESPONSE" which is created in previous section. This tag is used to copy input data into that structured data format.

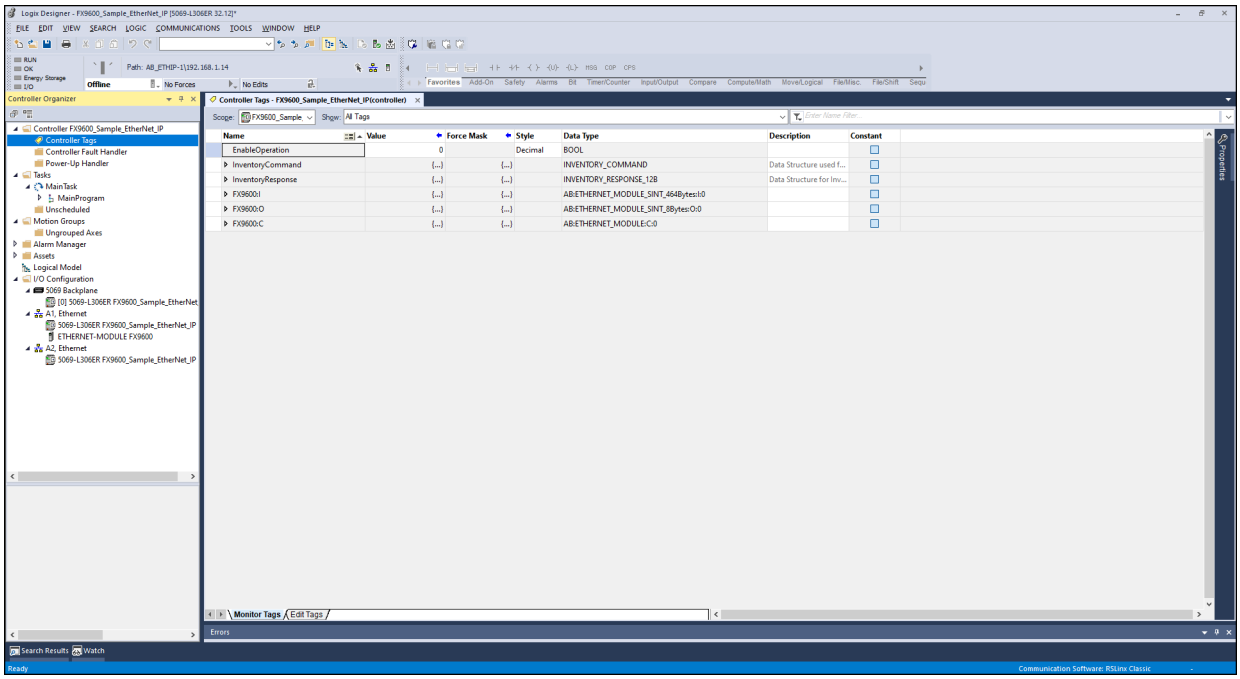
To create a controller tag:

1. Double-Click on **Controller Tags** under **Controller**.
2. Select the **Edit** tab at bottom of the right pane.



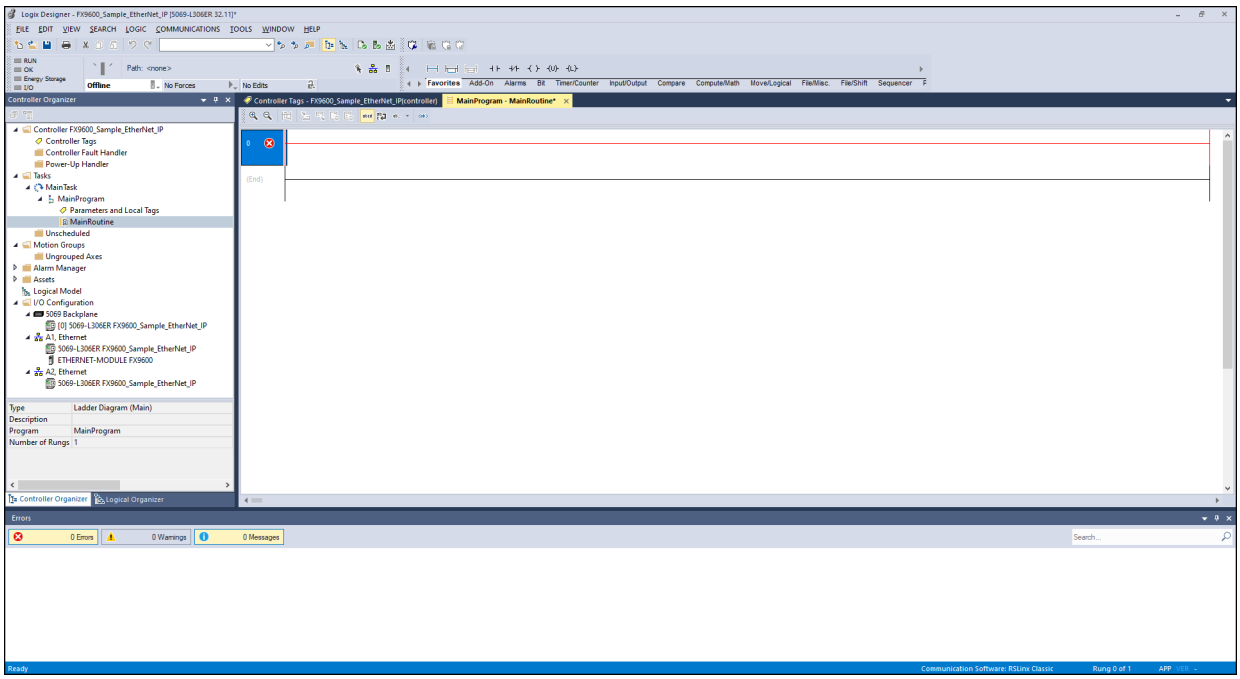
3. Click in the empty row as shown above and enter the tag name **`EnableOperation`**.
4. In the **Data Type** field, enter **`BOOL`**.
5. Press enter key to create the tag.
6. Click in the empty row as shown above and enter the tag name **`InventoryCommand`**.
7. In the **Data Type** field, enter **`INVENTORY_COMMAND`**.
8. Press enter key to create the tag.
9. Click in the empty row as shown above and enter the tag name **`InventoryResponse`**.
10. In the **Data Type** field, enter **`INVENTORY_RESPONSE`**.
11. Press enter key to create the tag.

Figure 32 Controller Tag



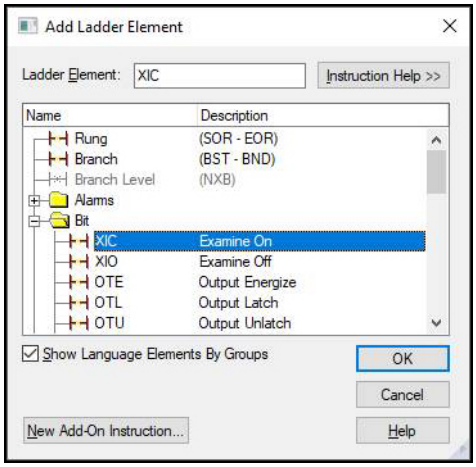
To Create a Ladder Logic:

1. From left pane, navigate to **Tasks > Main Task > Main Program > Main Routine**.

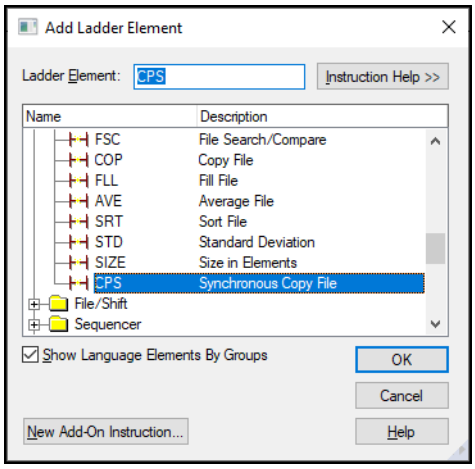


2. Double-click on **Main Routine**.

3. Right click on the rung and select **Add Ladder Element**. The **Add Ladder Element** window displays.



4. Navigate to **Bit** operators and select **(XIC) Examine On**.
5. Click **OK** to add the element.
6. Double-click on the **Examine On** element.
7. Enter `EnableOperation`.
8. Right click on the rung and select **Add Ladder Element**. The **Add ladder Element** window displays.

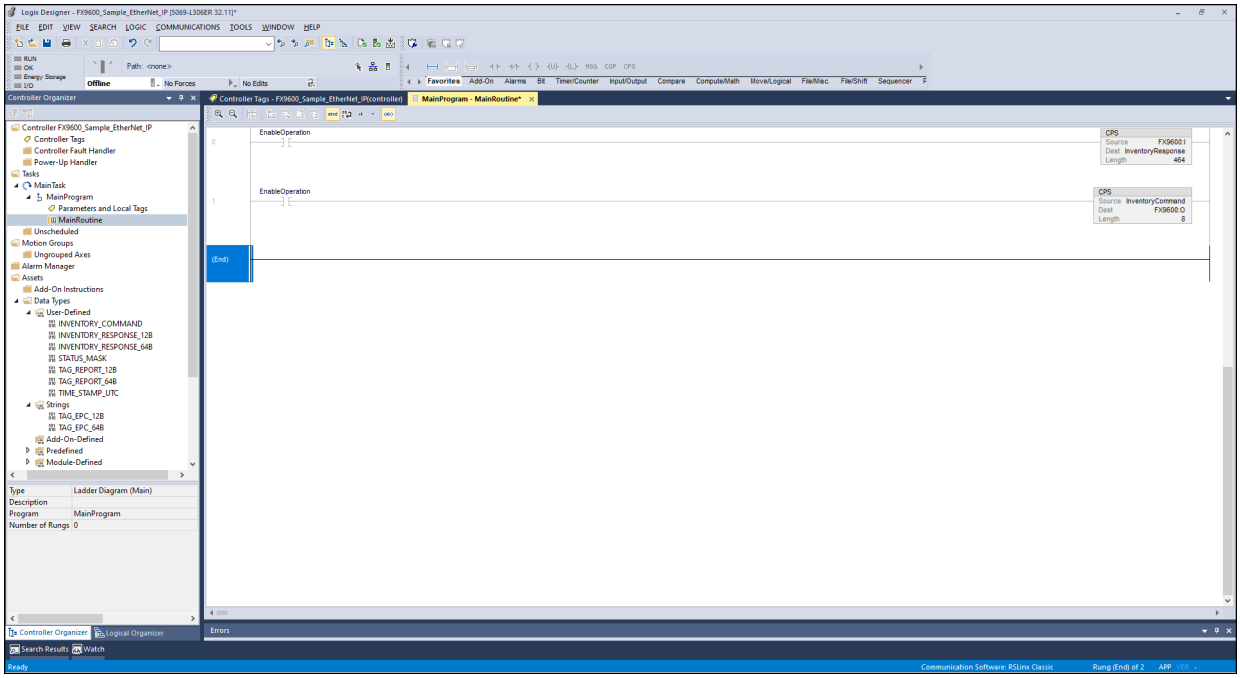


9. Navigate to **File/Shift** and select **(CPS) Synchronous Copy File**.
10. Click **OK** to add the element.
11. Choose the **Synchronous Copy File** block and set:
    - **Source**: `InventoryCommand`
    - **Dest**: `ModuleName:O`
    - **Length**: `8`.
12. Right-click the rung and select **Add Rung** to add a new rung.
13. Right-click on the rung and select **Add Ladder Element**.
14. Navigate to **Bit** operators and select **(XIC) Examine On**.
15. Click **OK** to add the element.
16. Double-click on the **Examine On** element.
17. Enter `EnableOperation`.

18. Right click on the rung and select **Add Ladder Element**.
19. Navigate to **File/Misc** and select **(CPS) Synchronous Copy File**.
20. Click **OK** to add the element.
21. Choose the **Synchronous Copy File** block and set:
    - **Source:** `ModuleName:I`
    - **Dest:** `InventoryResponse`

Length: `464`.

Figure 33 Ladder Logic



Loading the Project to the PLC Controller

After all previous steps are complete, the sample application project is ready to be loaded in the PLC Controller. To load the project to PLC, perform the following steps to configure the controller path in the Studio 5000 Logix Designer project.

Configure Driver for EtherNet/IP and Find Controller using RSLinx

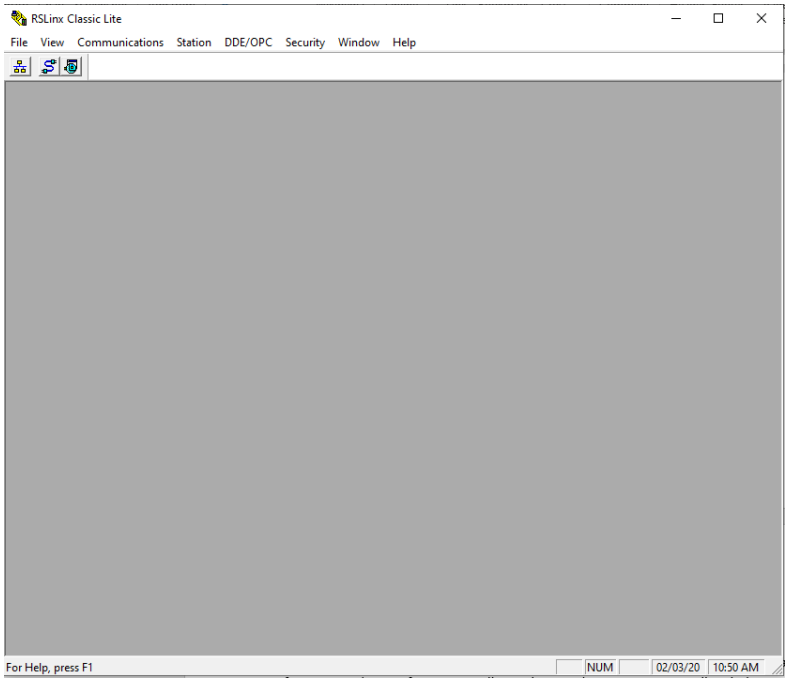
Before loading the project to PLC, the PLC Controller must be located over the LAN. To do this, use RSLinx Classic, which is part of the Studio 5000 Logix Designer installation. RSLinx Classic is used to find the PLC Controller over a network. To find your PLC Controller over the network make sure your PLC Controller is powered up and connected with the network through an Ethernet cable.

First, configure your PLC to assign an IP address. Generally, it is configured via USB. Use your PLC specific guide to configure the PLC Controller with an IP address.

When the PLC Controller is powered up and the IP address is assigned perform the following steps.

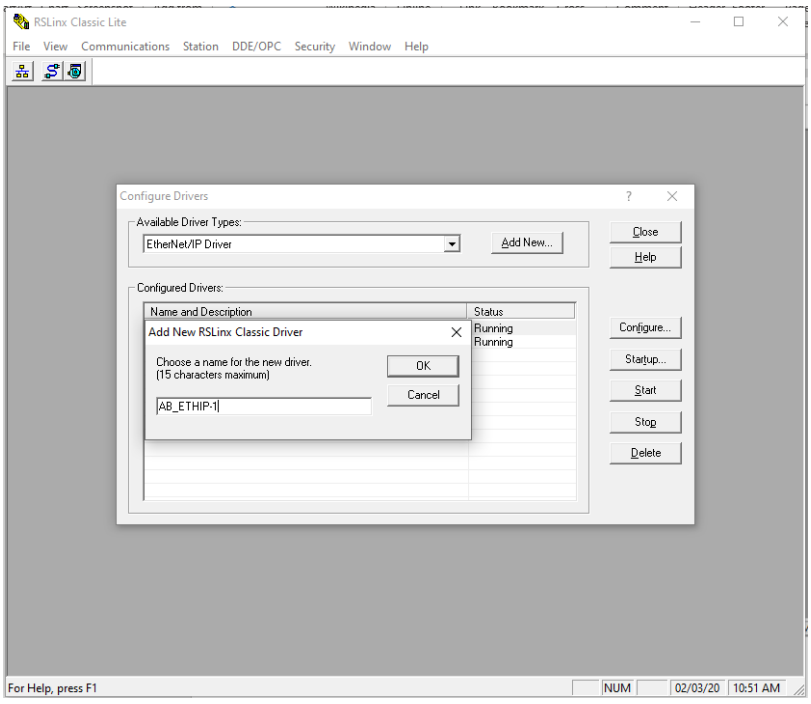
1. From the Windows Start menu, open RSLinx Classic. The RSLinx Classic window displays.

Figure 34 RSLinx Classic Window



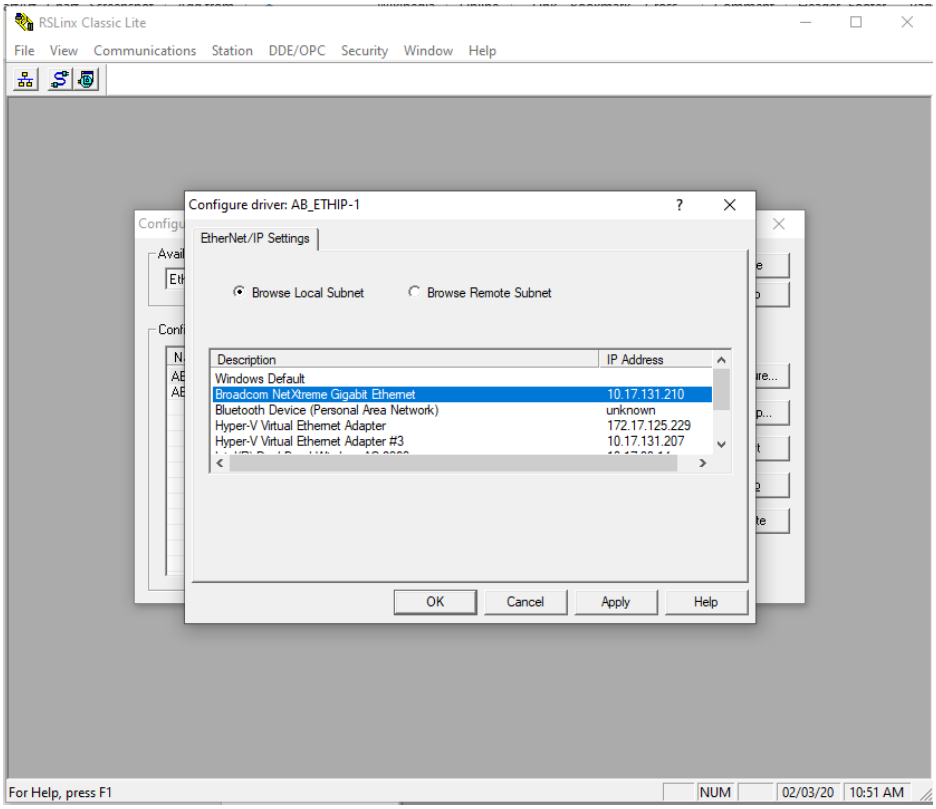
2. From the Communications menu, select Configure Drivers to display the Configure Drivers window.

Figure 35 Configure Drivers

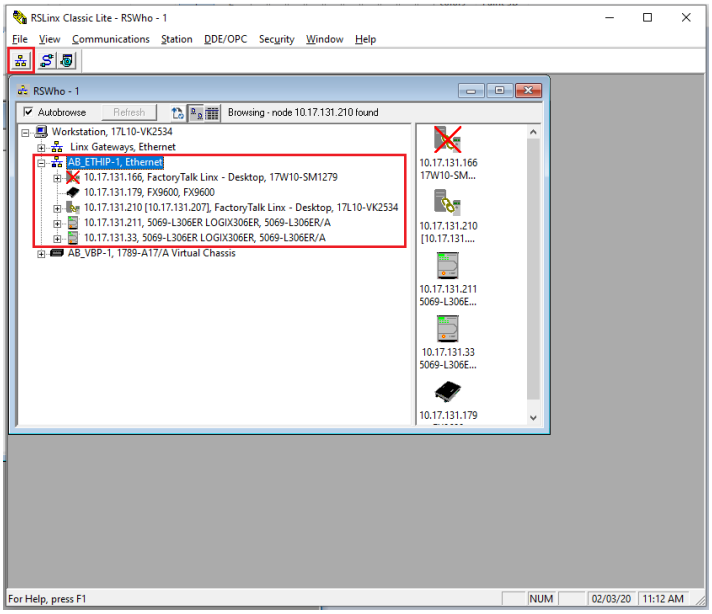


3. From the Available Driver Types drop-down list select EtherNet/IP Driver and click **Add New**.
4. In the New Driver window enter a name for the driver and click **OK**.
5. The new driver displays in the Configured Drivers space.
6. Select the newly created driver and click **Configure** to open the driver configuration dialogue.

Figure 36 Driver Configuration



7. Select the LAN and Ethernet interface to which the PLC Controller is attached.
8. Click **OK** to configure the driver interface.
9. Click **Start** to run the driver, if the driver status is not already running.
10. Close the Configuration window.
11. Click **RSWho** to display the **RSWho - 1** dialog box.



12. Expand the driver name (created in previous steps) to display all the EtherNet/IP capable devices.

13. Make sure your PLC Controller appears in the list. If it does not appear debug the driver configuration and Ethernet network.
14. The PLC Controller communication path is now set up and can be configured in the Studio 5000 Logix Designer project.

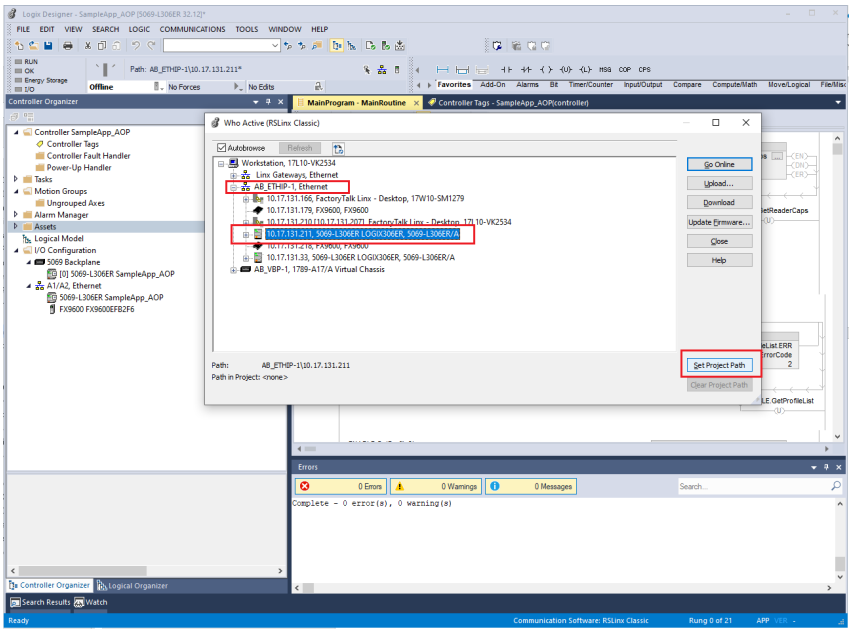
Set the Controller Path in Project

It is required to setup the PLC Controller path in the Studio 5000 Logix Designer to specify with which PLC Controller to communicate. Once the path is setup, the project can be loaded to the PLC Controller and the project can be controlled using the Studio 5000 Logix Designer by modifying data values in the project Controller Tags.

To set up the controller path in Studio 5000 Logix Designer.

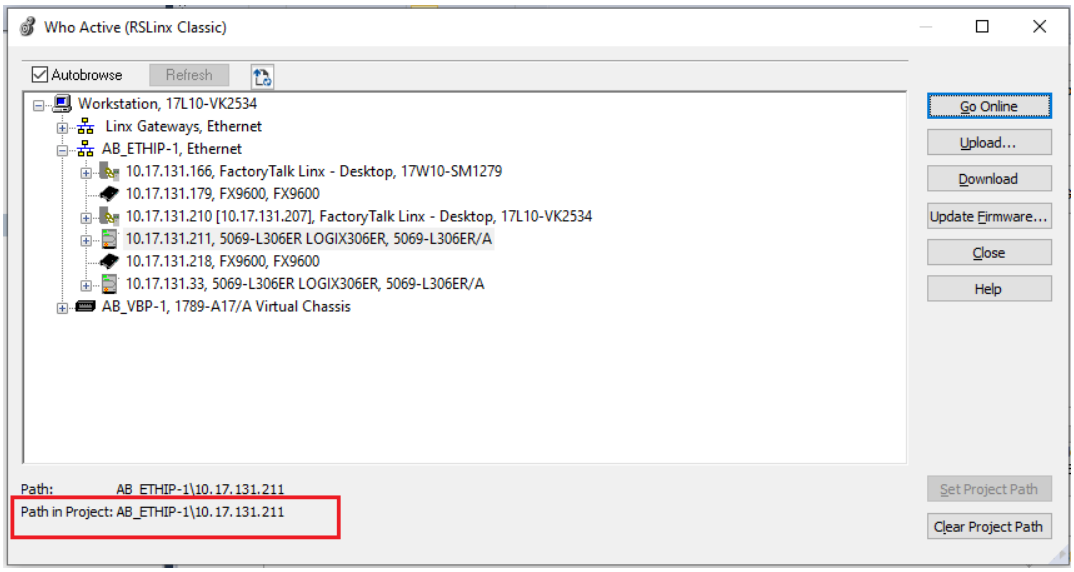
1. From the Studio 5000 Logix Designer project, select Who Active from the Communications menu to display the Who Active dialogue with configured drivers showing all available EtherNet/IP devices.

Figure 37 Who Active Dialogue



2. Expand the driver created in the section, Configure Driver for EtherNet/IP and Find Controller using RSLinx on page 39 and select the PLC Controller with which to communicate.
3. Click **Set Project Path**. The PLC Controller path is set as the path in the project with the driver name and IP address of the controller.

Figure 38 Project Path



4. Click **Close**. This sets the PLC Controller path in your current project.

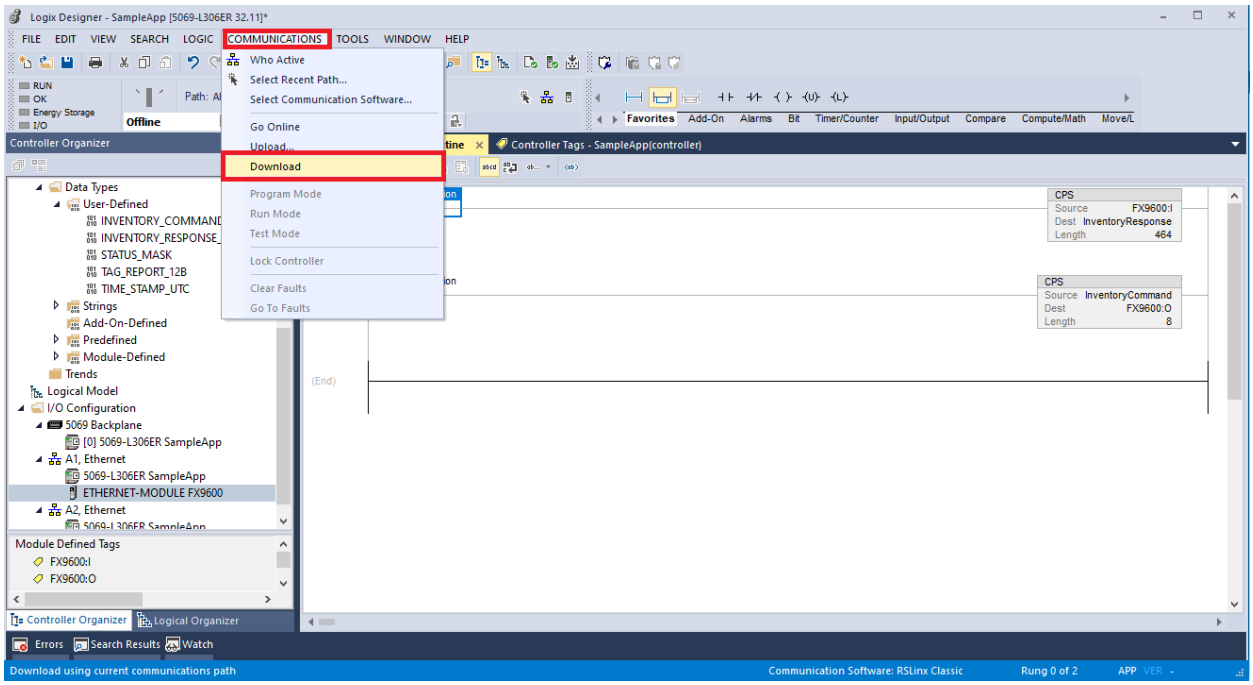
Load Project to PLC Controller

After configuring the EtherNet/IP driver, and locating and setting the controller path to Studio 5000 Logix Designer project, the project can be loaded to the PLC Controller. When the PLC Controller is loaded with the project, the PLC Controller mode can be changed to establish communication with the RFID reader.

To load the project to the PLC Controller:

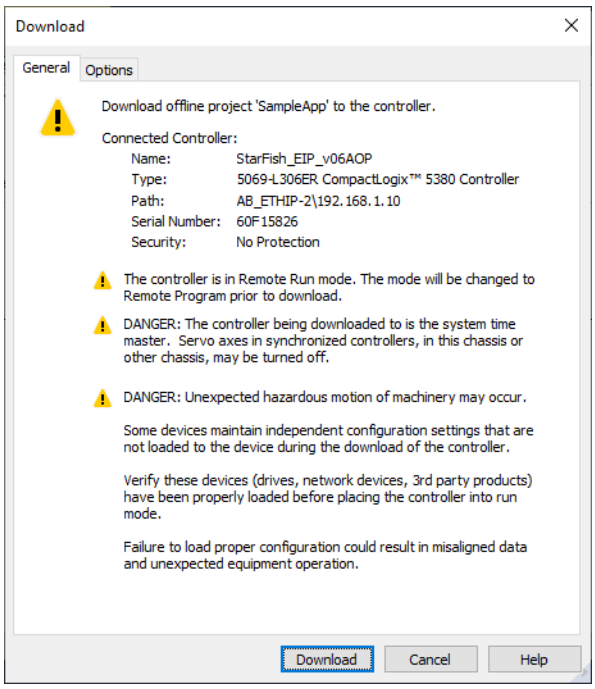
1. From the Studio 5000 Logix Designer project, select **Download** from the **COMMUNICATIONS** menu.

Figure 39 Communications > Dowload



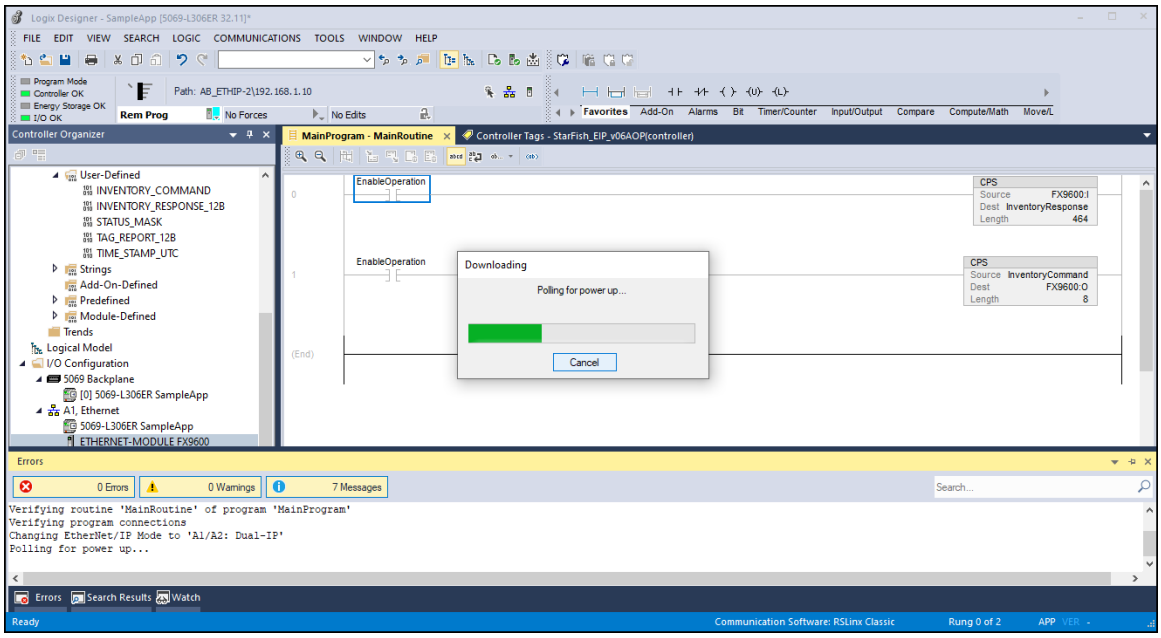
2. A Confirmation window displays with the controller information. Click **Download** to confirm.

Figure 40 Download Confirmation



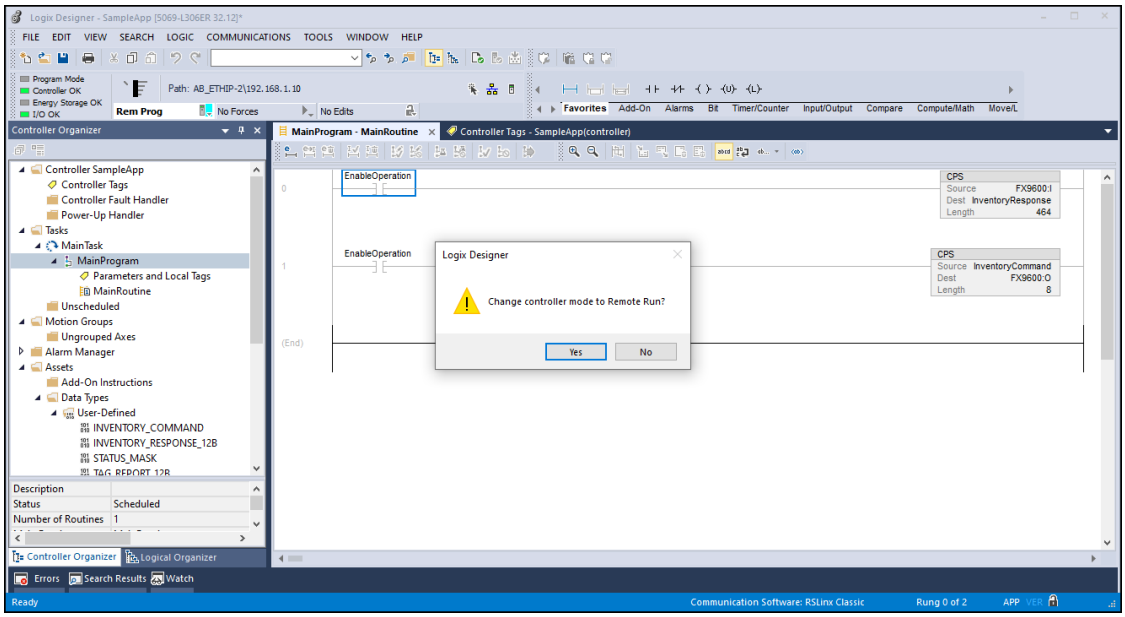
3. The project compiles and starts downloading to the PLC Controller.

Figure 41 Download Status



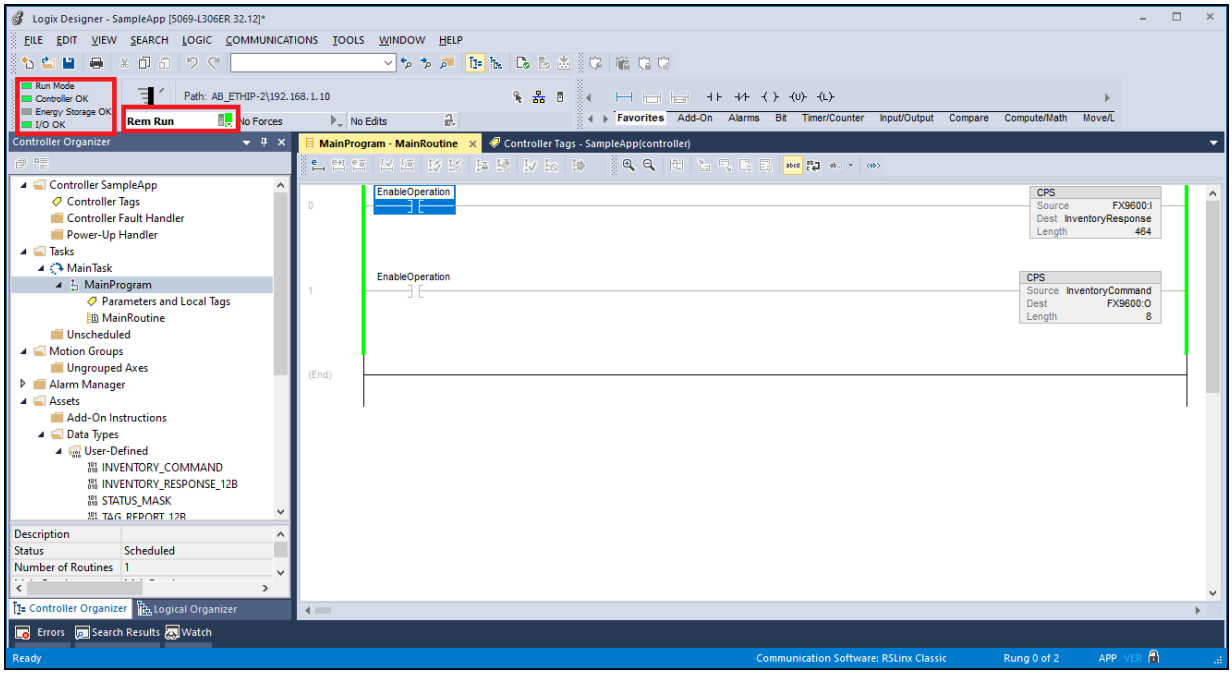
4. When the download is complete a prompt requires the user to change the controller mode to Run Mode. Click **Yes** to set the controller mode to Remote Run.

Figure 42 Set Remote Run



5. Studio 5000 Logix Designer displays green controller statuses.

Figure 43 Green Status Modes



At this stage, the project is loaded to the PLC Controller and running and the command is ready to be sent to the RFID reader using Studio 5000 Logix Designer.

Before sending the command to the RFID reader, the reader must start the application to make the connection. Reading Reader Capabilities on page 46 explains how to set up the reader with the EtherNet/IP application.

RFID Configuration and Operations Using the EtherNet/IP (EIP) Sample App

Reading Reader Capabilities

When the project is loaded to PLC Controller, and the PLC Controller is in Run Mode, RFID reader communication can begin and RFID operations can be performed.

Use Rung 0 to read the reader capabilities. Figure 44 shows an explanation of the instructions defined in Rungs to read the Reader Capabilities.

Figure 44 Rung Instructions for Reading Capabilities

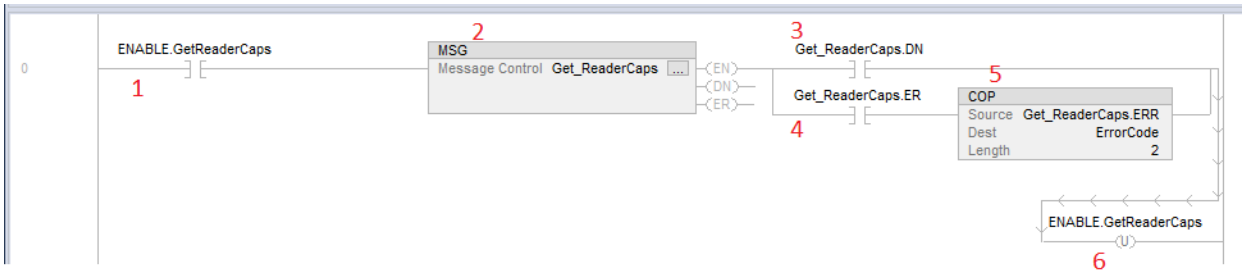


Table 1 Rung Instructions

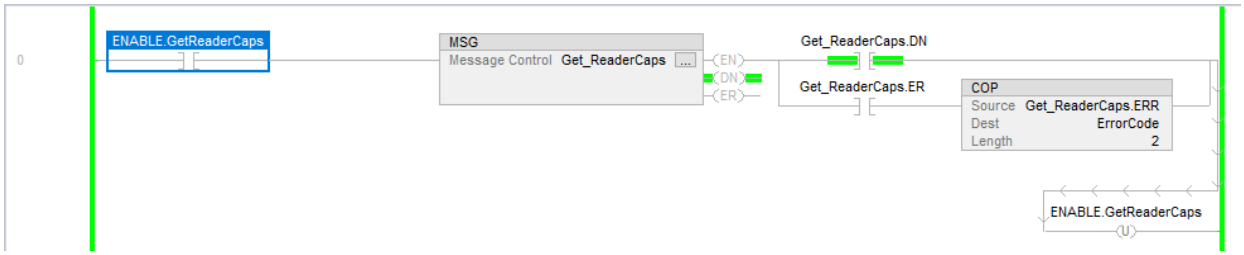
Number	Description
1	This is the Examine instruction to enable the Rung. It examines on the ENABLE.GetReaderCaps flag defined in the common ENABLE structure for all the flags. Once this flag is TRUE, the Rung becomes enabled and it executes the next instruction.
2	MSG instruction is to perform explicit operation with reader over EIP protocol. This uses the controller Tag Get_ReaderCaps of type MESSAGE. This instruction is configured for Explicit Message 0x64 (Reader Capabilities) to perform Get Attribute service request. The data read from the instruction is copied in the ReaderCapabilities Tag created under controller Tags. There is branching in this Rung after the MSG instruction to get the error code if there is a failure to get the Reader Capabilities.
3	This Examine On Instruction examines the done flag of the Get_ReaderCaps MSG instruction. Once the instruction is successfully executed, it enables to execute the next instruction.

Table 1 Rung Instructions (Continued)

Number	Description
4	This Examine On Instruction examines on the error flag of the Get_ReaderCaps MSG instruction. If the MSG instruction fails to read the Reader Capabilities, this is enabled and the next instruction in the branch executes.
5	This is the Copy instruction to cope with the error code from the Get_ReaderCaps instruction to ErrorCode Tag.
6	This last instruction is Output Unlatch. Once the Rung execution is done, this unlatches the ENABLE.GetReaderCaps and the Rung is disabled.

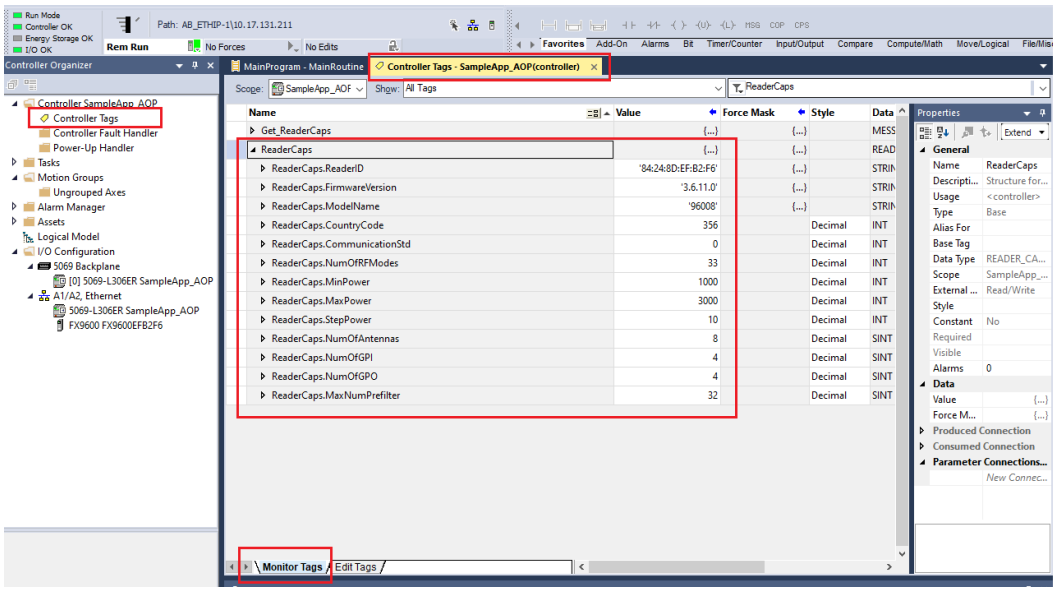
To execute the Reader Capabilities explicit message, select Instruction 1 (shown in Figure 44) and press Ctrl + T to toggle the ENABLE.GetReaderCaps flag to TRUE. The Rung enables to get the Reader Capabilities.

Figure 45 ENABLE.GetReaderCaps flag = TRUE



The output can be viewed under Controller Tags. To see the output, select Controller Tags from the left pane and filter Tags with ReaderCaps. This displays the Reader Capability Tag. Expand the Tag and it lists the data read from reader.

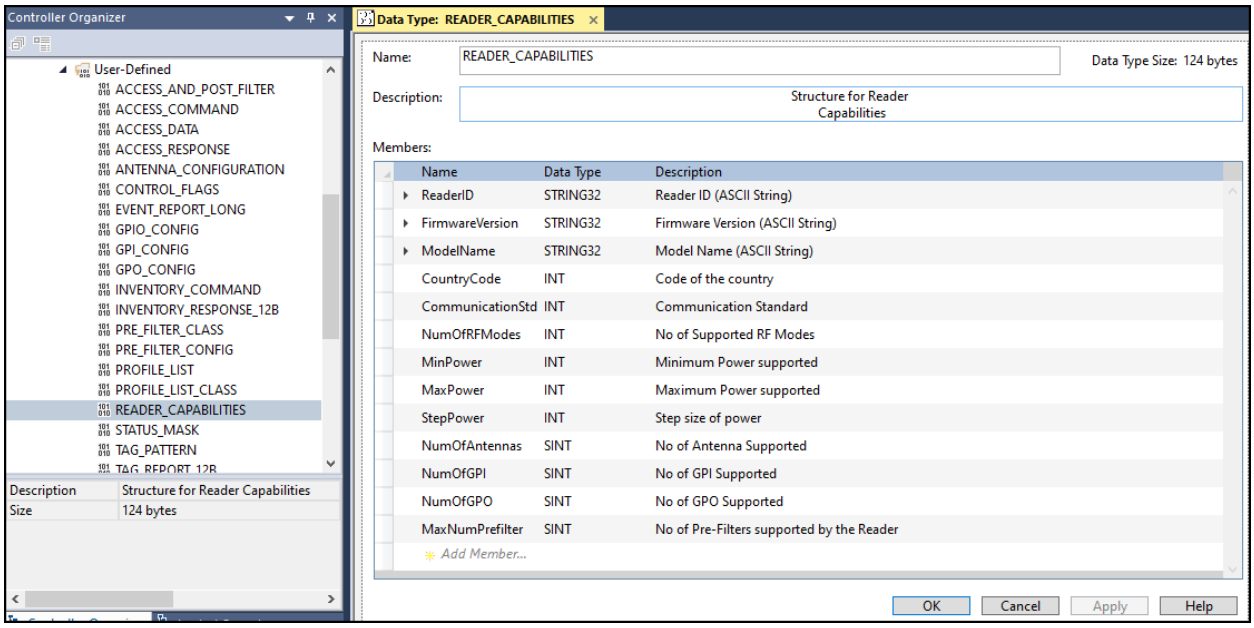
Figure 46 ReaderCaps Data



Reader Capabilities Parameters

- Vendor Class: 64

- Data type size: 124 bytes.



Customize Configuration for RFID Operation

There are additional configuration options for the reader. These configurations can be applied using the explicit messaging option to the reader.

This section describes various configuration options and how to use them. Detailed information about configuration messages and their configuration fields is defined in Zebra EtherNet/IP Data Model document.

NOTE: The Zebra EtherNet/IP Data Model document is included in the EtherNet/IP deliverable and is also available at: zebra.com/support.

Reader Profile

Zebra RFID readers have pre-installed default profiles. The Reader Profile provides the default configuration for different use cases. Through the EtherNet/IP application, the user can get the pre-installed profile names and can choose to activate the desired profile.

Profile List explicit message is constructed as shown in the data model in Table 2.

Table 2 Profile List Explicit Message Data Model

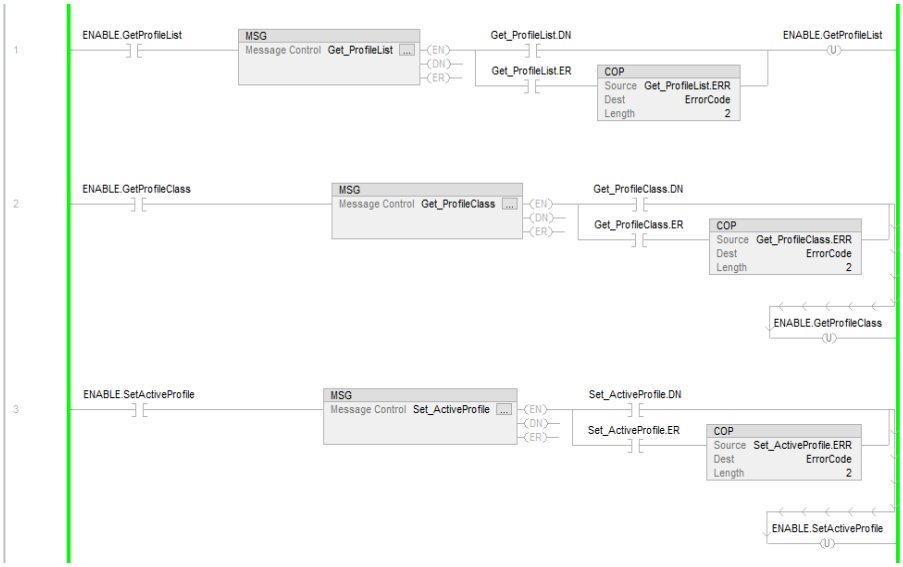
Instance	Attribute ID	Name	CIP Data Type	Data Value	Access Rule
Class (Instance 0)	1	Revision	UINT	1	Get
	2	Max Instance	UINT	32	Get
	3	Num Instances	UINT	Varies	Get
	100	Active Profile Instance	UINT	Varies	Get/Set
1	1	Profile Name	STRING64	Varies	Get
2	1	Profile Name	STRING64	Varies	

Instance 0 has the following members:

- Revision: EtherNet/IP standard for version information.
- Max Instance: Maximum number for instances this explicit message can support. Here this represents the maximum number of profiles the can be supported by model.
- Num Instances: This represents the currently available number for instances. Here it represents the number of profiles currently installed on reader.
- Active Profile Instance: This attribute represents the currently active profile Instance. The user can perform a Get operation on the profile list on that specific instance to read the name of the currently active profile.
- Profile Name: There are multiple rows for profile name, the number of row/instances depends the number of available profiles in reader. The user can perform a Get operation to get the name of a profile on a specific instance.

Figure 47 shows the Rungs created to perform an operation on reader profiles.

Figure 47 Rungs Performing an Operation

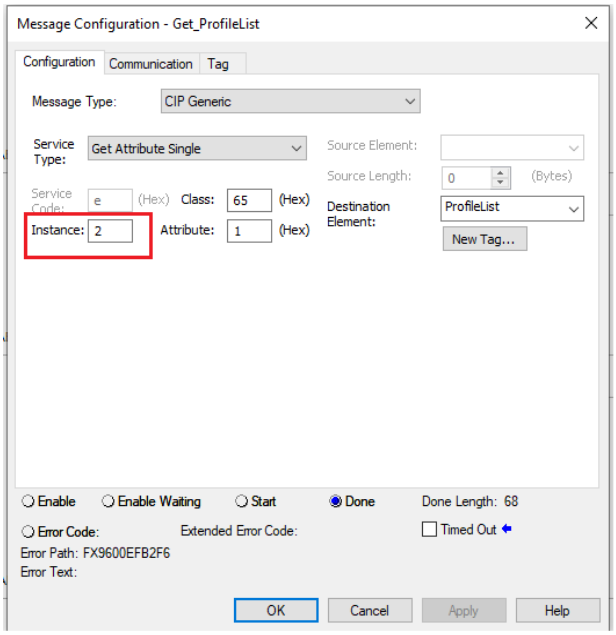


From the screen in Figure 47, the user can read the name of a profile by providing the index of the profile.

To specify the index of the profile name to read:

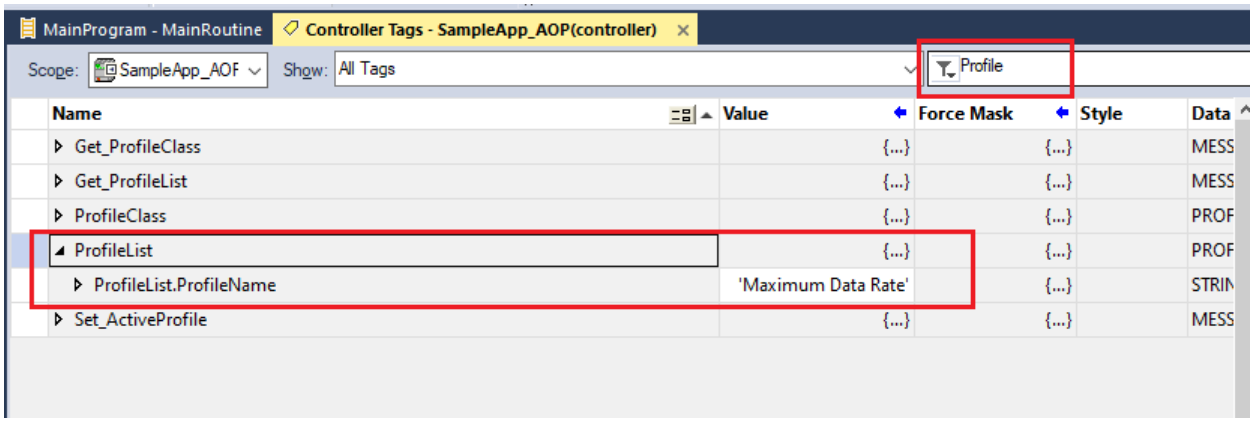
1. Click on the configuration for MSG instruction on Rung 1 and specify the index number in the Instance field as shown in Figure 48.

Figure 48 Specifying the Profile Index Number



2. Click **OK** after inputting the Instance number.
3. Press Ctrl + T keys to toggle the Examine on ENABLE.GetProfileList instruction in Rung 1. It performs the operation to read thee profile name of the provided Instance.
4. Read the output from ProfileList Tag in the controller Tags window.

Figure 49 Controller Tags

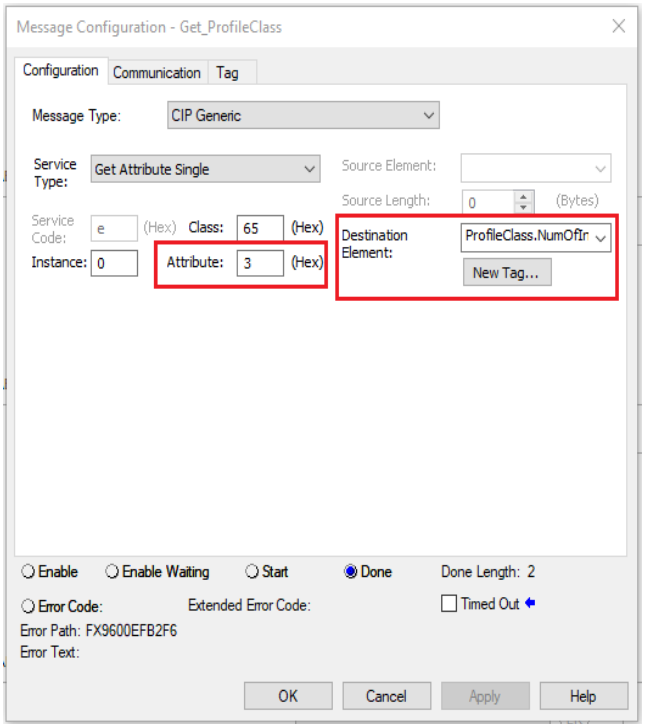


5. To get the information about how many profiles are available in the reader and to know which is the currently active profile, the user should perform explicit message on the Profile List Class. Rung 2 was created to demonstrate this.

   a. Click on the configuration for the MSG instruction in Rung 2 and change the attribute value to 3 (Num Instances).

   b. Select the NumOfInstances field from the ProfileClass structure in the Destination Element field.

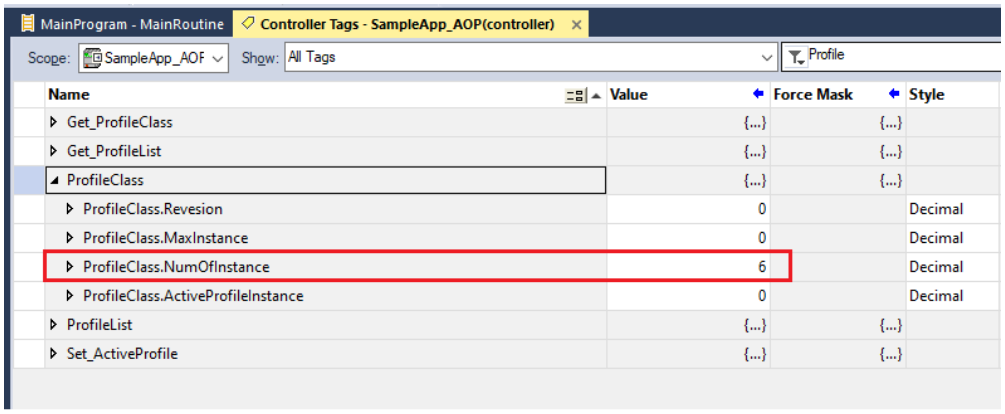
Figure 50 Obtaining the Number of Profiles



   c. Click **OK**.

   d. Perform the Get operation on the Profile List Class in Rung 2 by toggling the ENABLE.GetProfileClass Examine On Instruction. When complete, the number of available profiles is read in the NumOfInstances variable in the ProfileClass structure.

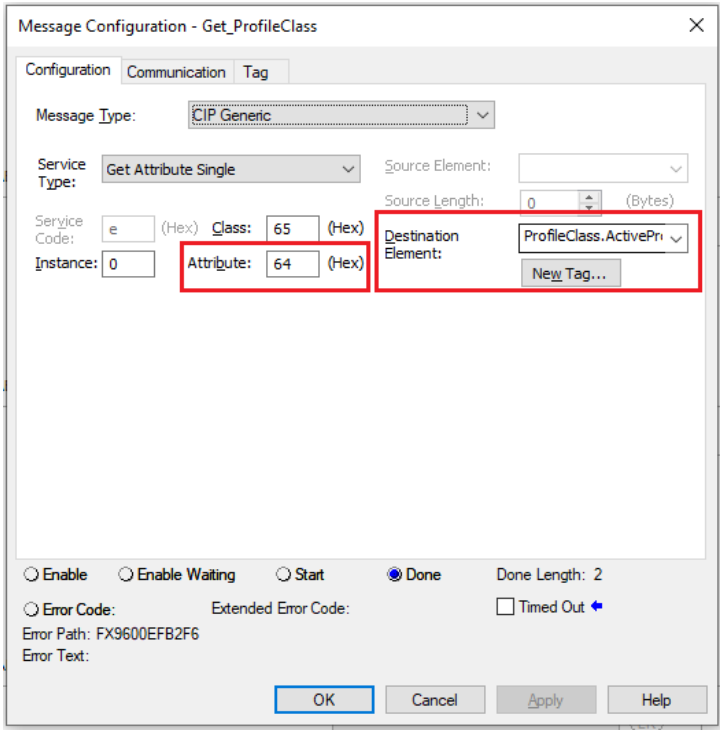
Figure 51 Number of Available Profiles



or

e. The user can also read the currently active profile instance by reading the Active Profile Instance from the Profile List class. To read the active profile index, configure the MSG instruction in Rung 2 by changing the attribute to 100 (64 in hex) and the Destination Element to ProfileClass.ActiveProfileInstance.

Figure 52 Active Profile Instance from Profile List Class

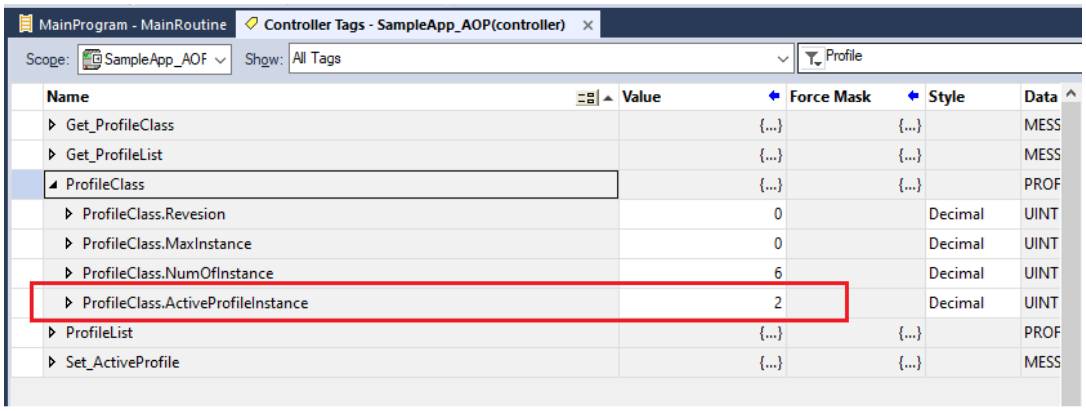


i. Click **OK**.

ii. Perform the Get operation on the Profile List class in Rung 2 by toggling the ENABLE.GetProfileClass Examine On Instruction.

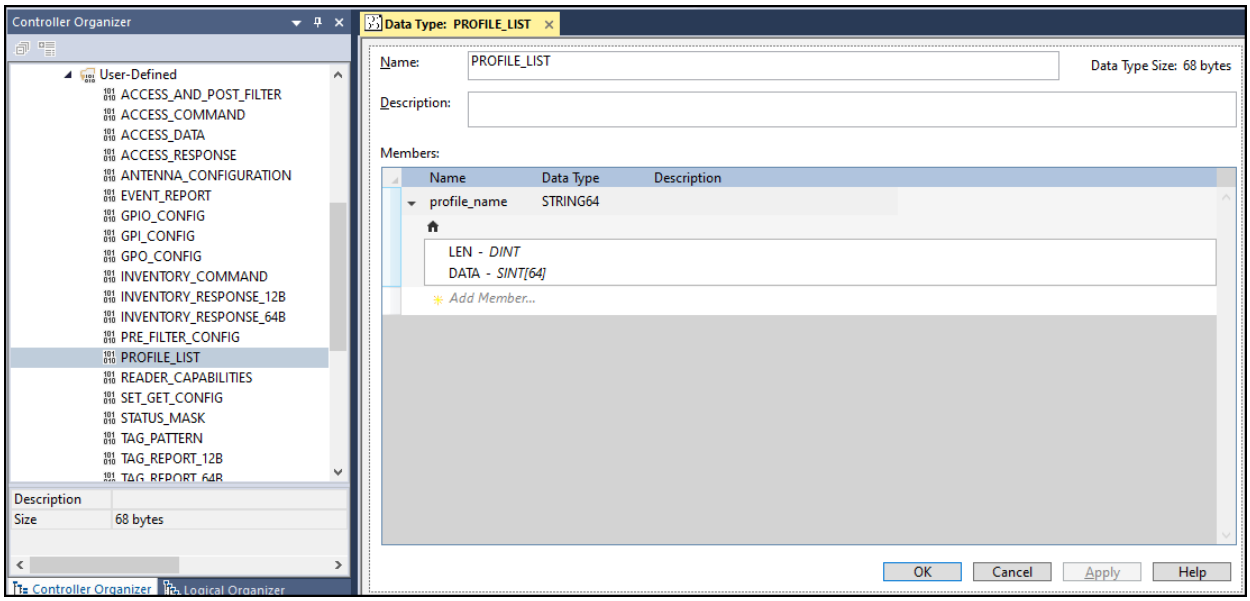
iii. When complete, the Active Profile instance is available in the ActiveProfileInstance field of ProfileClass structure.

Figure 53 Active Profile Instance



Profile List Parameters:

- Vendor class: 65
- Data type size: 68 bytes.



Changing the Active Profile in the Reader

To change the Active Profile in the reader the user can perform Set Operation on the Active Profile Instance variable in the profile list explicit message. Rung 3 was created to demonstrate the functionality in sample application.

To change the active profile in reader:

1. Specify the desired profile index to be activated in the ActiveProfileInstance field of the ProfileClass structure.
2. Once specified, go to the Main Routine tab and perform the Set_ActiveProfile MSG instruction in Rung 3 by toggling ENABLE.SetActiveProfile Examine On Instruction. Set_ActiveProfile MSG instruction in Rung 3 is pre-configured to perform the Set Active Profile operation.

Antenna Configuration

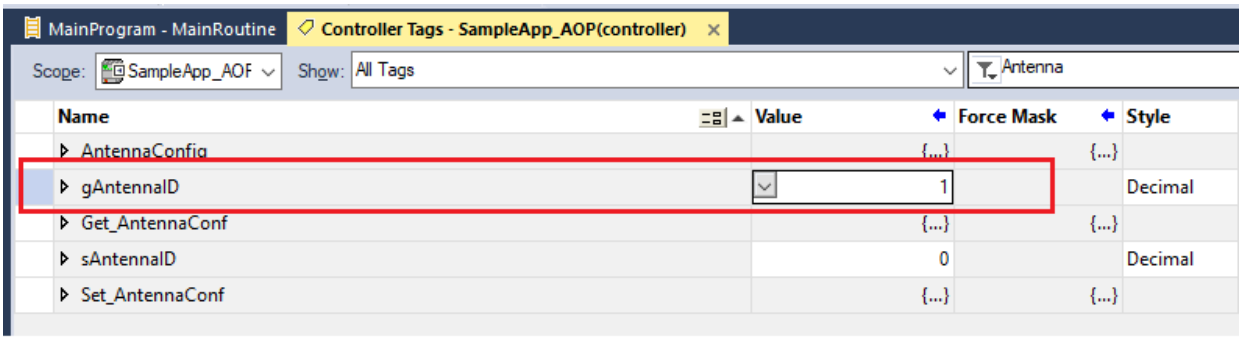
The sample application can also configure reader antennas. Figure 54 shows how to get and set the antenna configuration in Rung 4 and Rung 5.

Figure 54 Get and Set Antenna Configuration

To get the antenna configuration:

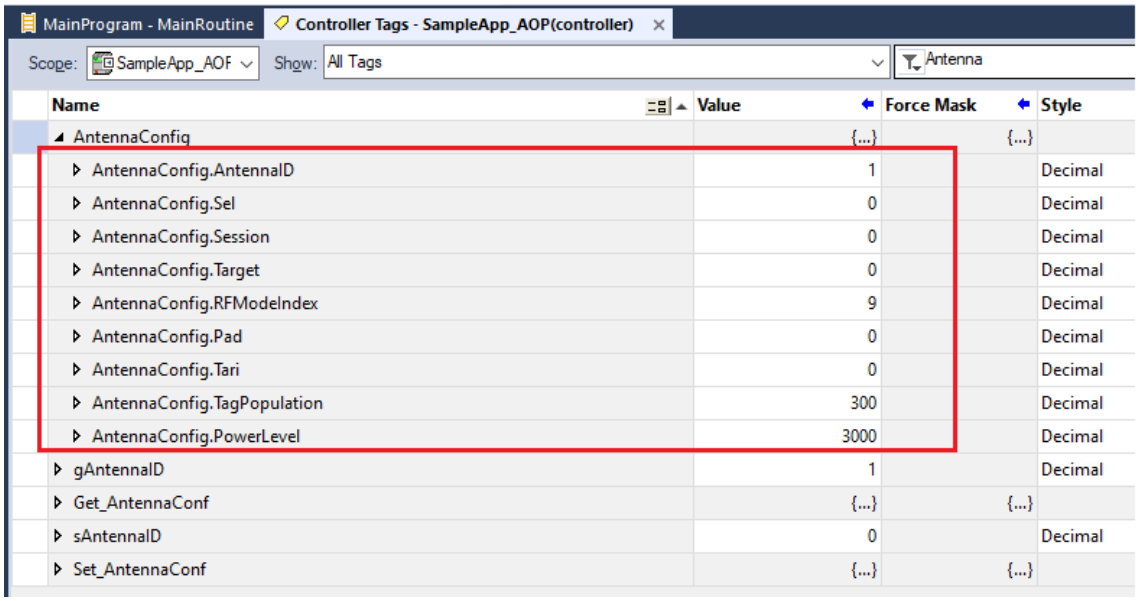
1. Specify the Antenna ID in the gAntennaID controller tag.

Figure 55 Specify Antenna ID



2. Enter the Antenna ID.
3. Select the **MainProgram - MainRoutine** tab.
4. Toggle the Rung 4 Examine On Instruction for ENABLE.GetAntennaConf. When enabled, the Get command is performed on the antenna configuration. Configuration values are read in the AntennaConfig structure.

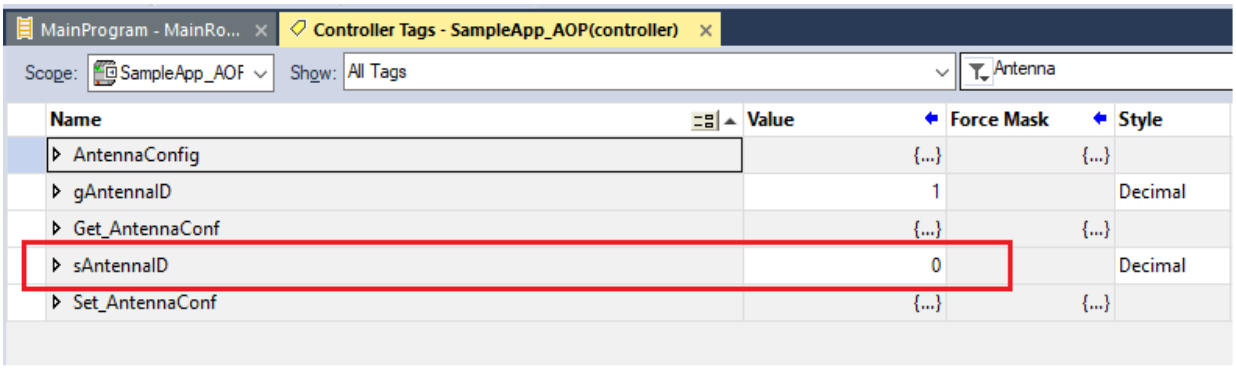
Figure 56 Antenna Configuration Values



To modify the antenna configuration in the reader:

1. Specify the desired configuration in AntennaConfig controller tag.
2. Enter the Antenna ID to apply the updated configuration. The Antenna ID must be provided in the sAntennaID controller tag.

Figure 57 Antenna ID to Modify



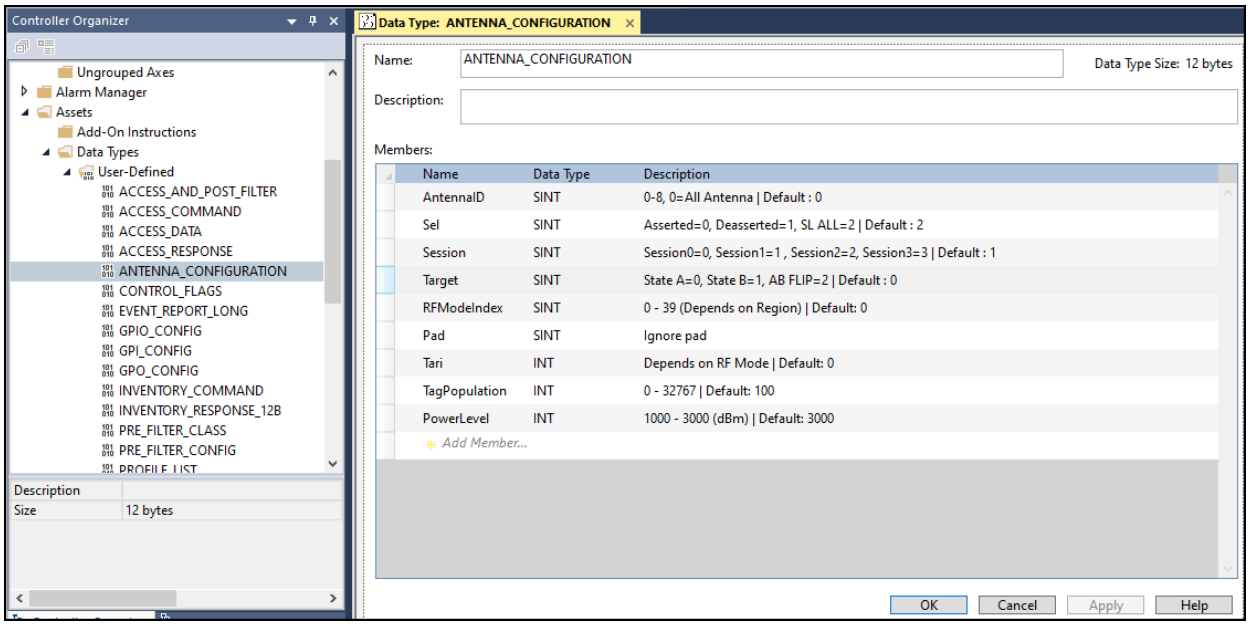
3. Select the **MainProgram - MainRoutine** tab.
4. Toggle the Rung 5 Examine On Instruction for ENABLE.SetAntennaConf. When enabled, the Set command is performed on the antenna configuration. Configuration values specified in the AntennaConfig tag are applied to the specified Antenna ID.

NOTE: Antenna ID can be 0 to set the same configuration on all antennas at once; but to get the configuration user need to provide specific antenna ID.

Antenna configuration parameters:

- Vendor class: 66

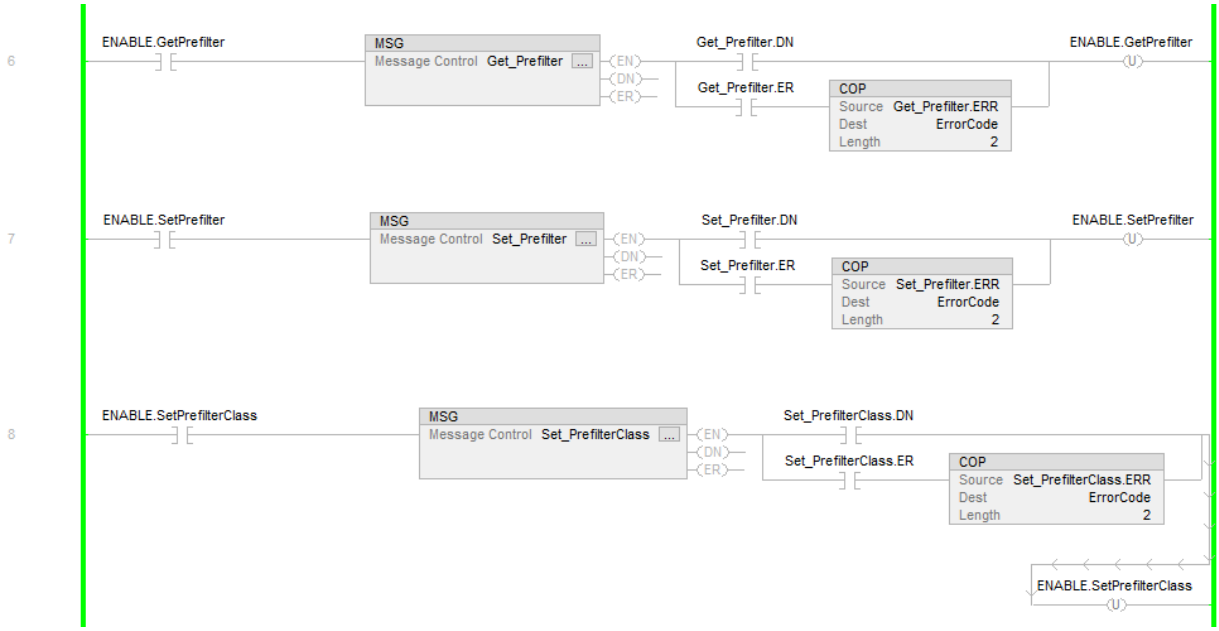
- Data type size: 12 bytes



Configuring Pre-Filter

This section explains how to add/delete and read pre-filters from reader. The maximum number of pre-filters the reader can support is 32. Each pre-filter is accessed with a specific pre-filter index as an instance number from the Pre-Filter explicit message defined in the Data Model. In Figure 58, Rung 6, Rung 7, and Rung 8 were created to demonstrate the use of the pre-filter explicit message.

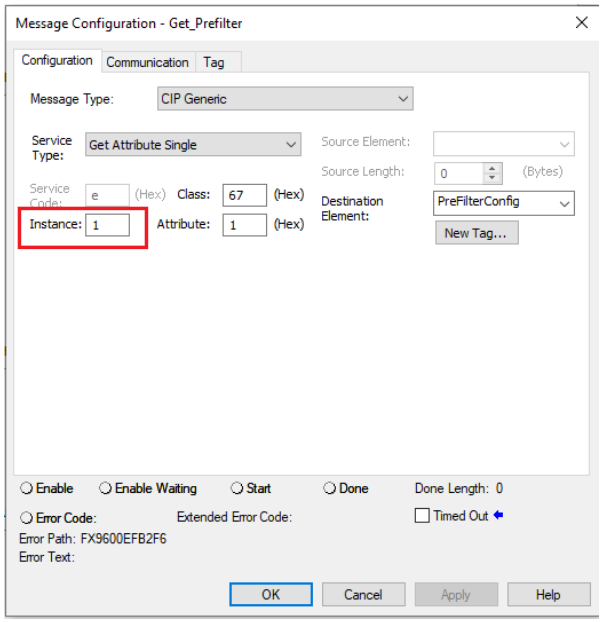
Figure 58 Pre-filter Explicit Message



To get the pre-filter from a specific index:

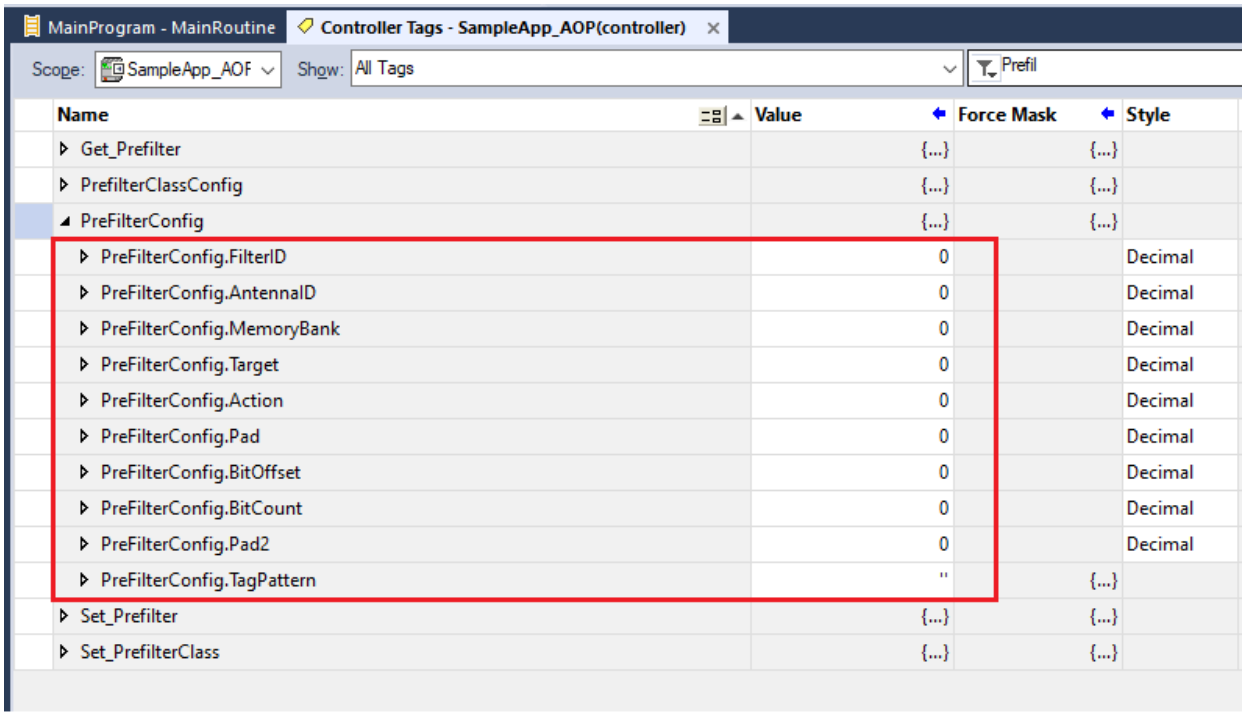
1. Specify the filter index as an Instance ID in the MSG configuration in Rung 6.

Figure 59 Specify the Filter Index



2. Enable Rung 6 by toggling ENABLE.GetPrefilter Examine On Instruction. The pre-filter configuration is read in the PreFilterConfig tag. (By default, there is no pre-filter added so all the values are 0.)

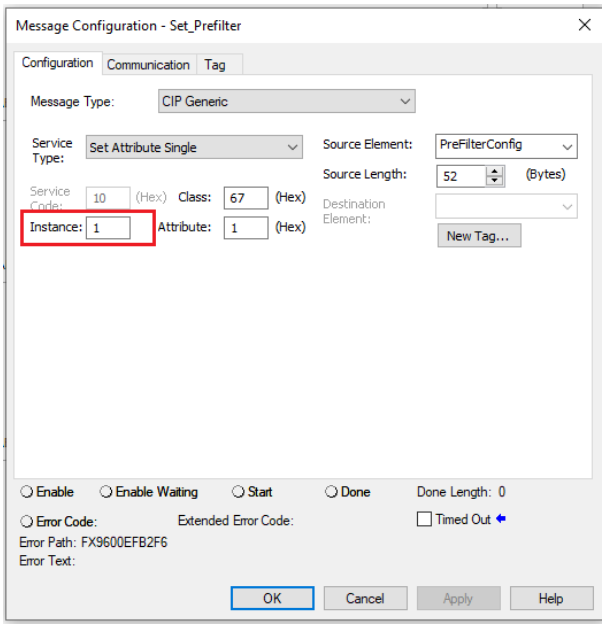
Figure 60 PreFilterConfig Tag



To add a pre-filter or modify the existing pre-filter on the specific index:

1. Modify the pre-filter values in the PreFilterConfig tag.
2. Select the MainProgram - MainRoutine tab.
3. Modify the MSG instruction in Rung 7 to specify the pre-filter index as an Instance on which the specified pre-filter needs to be added/modified.

Figure 61 Modify the MSG Instruction



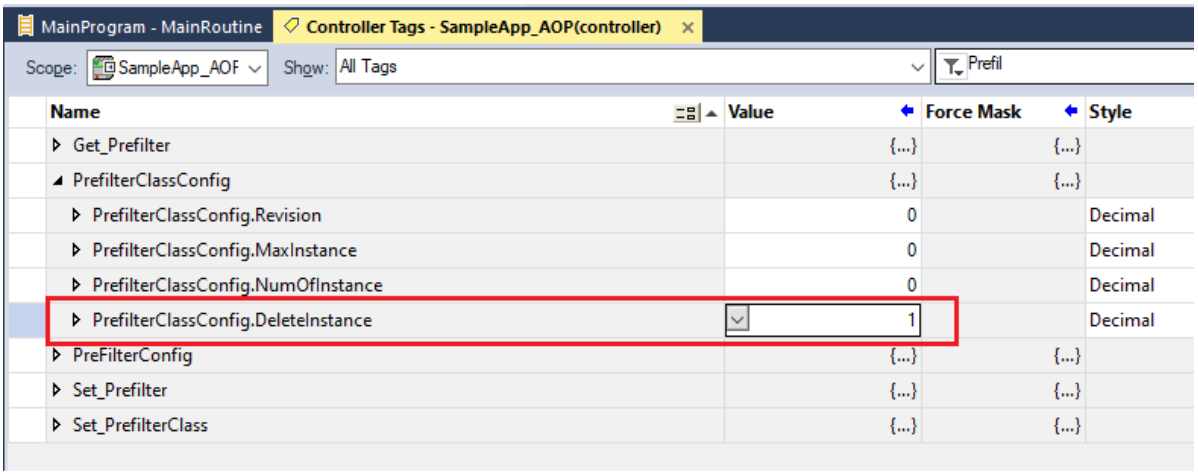
4. After specifying the pre-filter index to add/modify the pre-filter, enable Rung 7 by toggling ENABLE.SetPrefilter Examine On Instruction. Pre-filter configuration is added in the reader.

   The user can verify the values by performing GetPrefilter again on the same instance from Rung 6, as described above.

To delete an added pre-filter at a specific index:

1. Modify Rung 8 in the sample application with DeleteInstance from PrefilterClassConfig.
2. Specify the pre-filter index in the DeleteInstance field of the PrefilterClassConfig.

Figure 62 Delete a Pre-filter

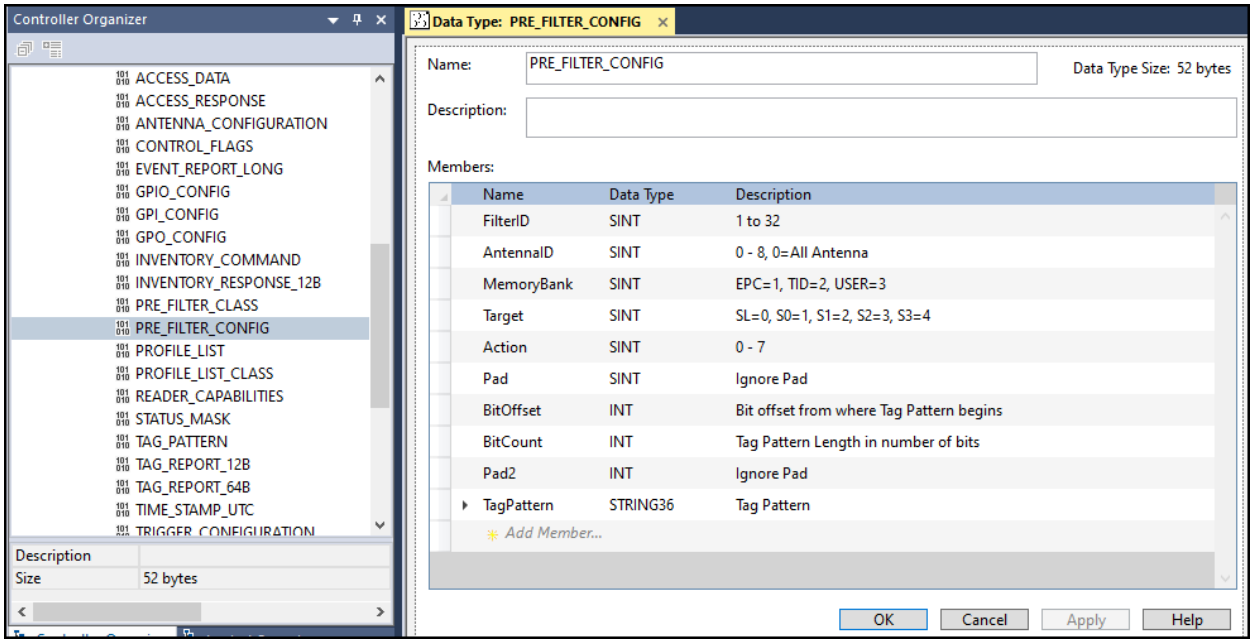


3. After specifying the pre-filter index to be deleted, select the MainProgram - MainRoutine tab and enable Rung 8 by toggling ENABLE.SetPrefilterClass Examine On Instruction.
4. This delete the specified pre-filter from the reader.

NOTE: To delete all pre-filters, use 0 in DeleteInstance field. This deletes all pre-filters from the reader.

Pre-Filter Configuration Parameters:

- Vendor class: 67
- Data type size: 52 bytes.

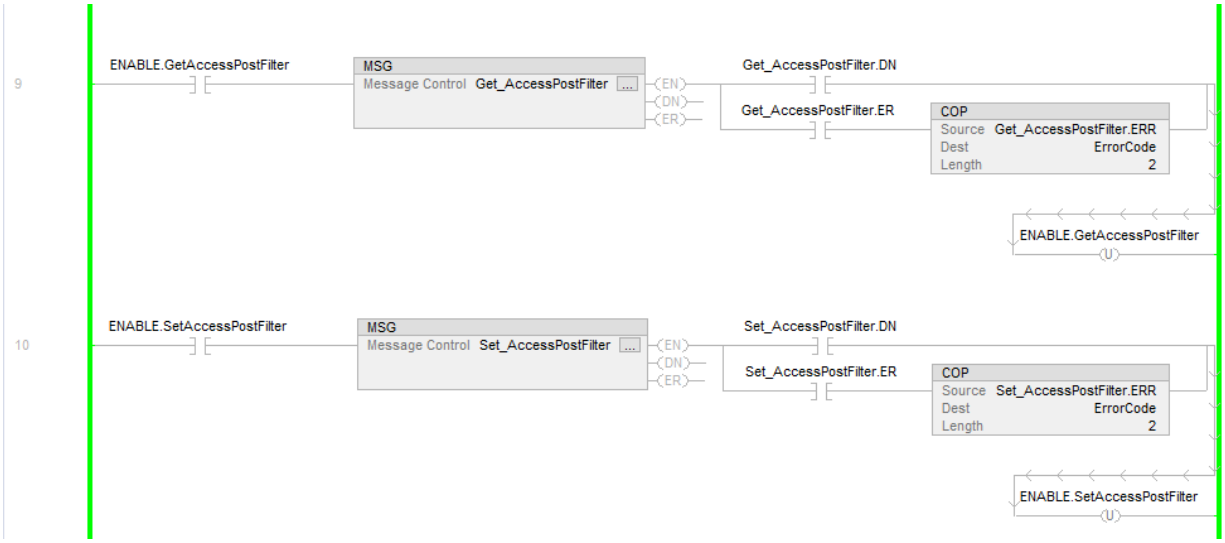


Configuring Access/Post-Filter

The Zebra RFID reader uses post-filter to filter tags that are read by the RFID radio engine. Rung 9 and Rung 10 were created to demonstrate how to configure post-filter.

This filter configuration can also be used with Access Filter when performing the Access operation (see Performing Access Operation on page 69).

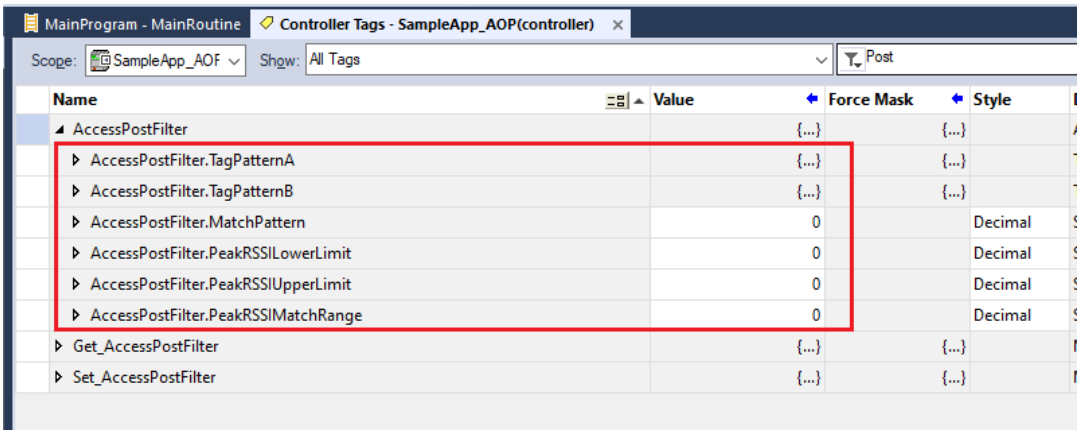
Figure 63 AccessPostFilter Operation



To get the value for the configured post/access filter:

1. Enable Rung 9 by toggling ENABLE.GetAccessPostFilter Examine On Instruction. This reads the filter configuration values in the AccessPostFilter tag in controller tags. By default, there is no Access/Post-filter so all the values are 0.

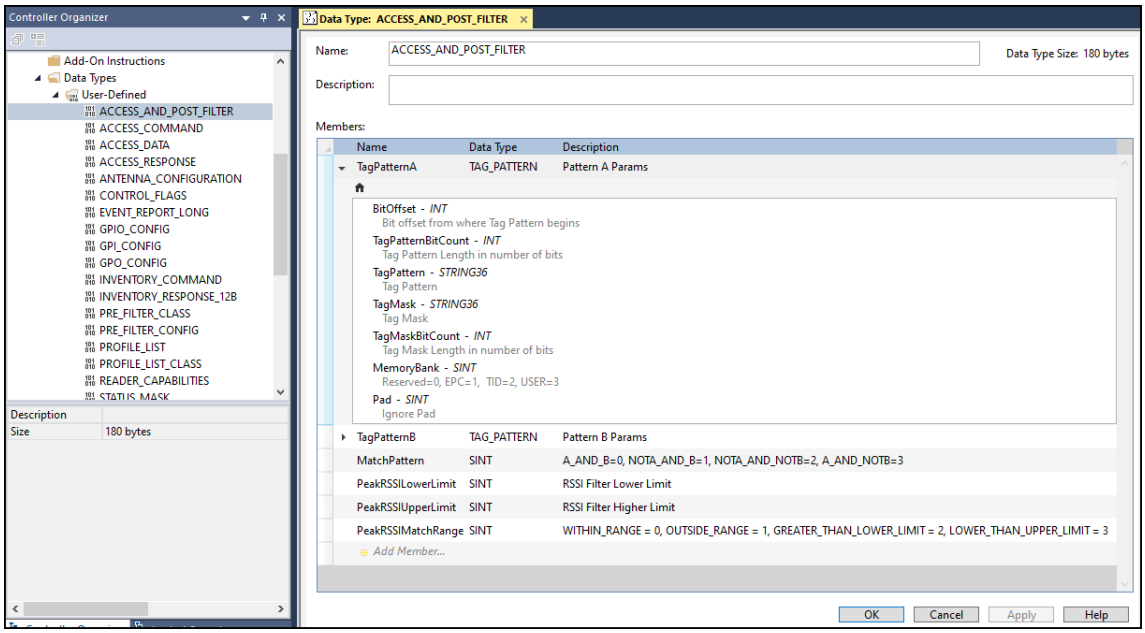
Figure 64 AccessPostFilter Tag



2. Access/Post-filter as per the requirement by modifying the values in the AccessPostFilter tag.
3. Select the **MainProgram - MainRoutine** tab and enable Rung 10 by toggling ENABLE.SetAccessPostFilter Examine On Instruction. This applies the post-filter configuration in the reader.
4. To verify, perform GetAccessPostFilter to validate the values.

Access/Post-Filter Configuration Parameters:

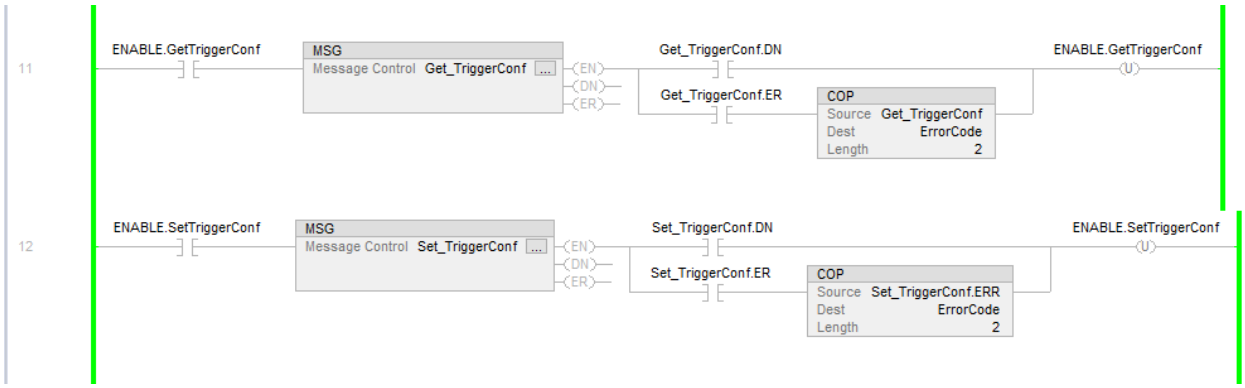
- Vendor class: 68
- Data type size: 180 bytes



Trigger Configuration

The Zebra RFID reader provides the mechanism to start and stop the trigger for the Inventory operation. These triggers can also be configured using the EtherNet/IP sample application. Rung 11 and Rung 12 were created to demonstrate configuring triggers values.

Figure 65 Configure Trigger Values

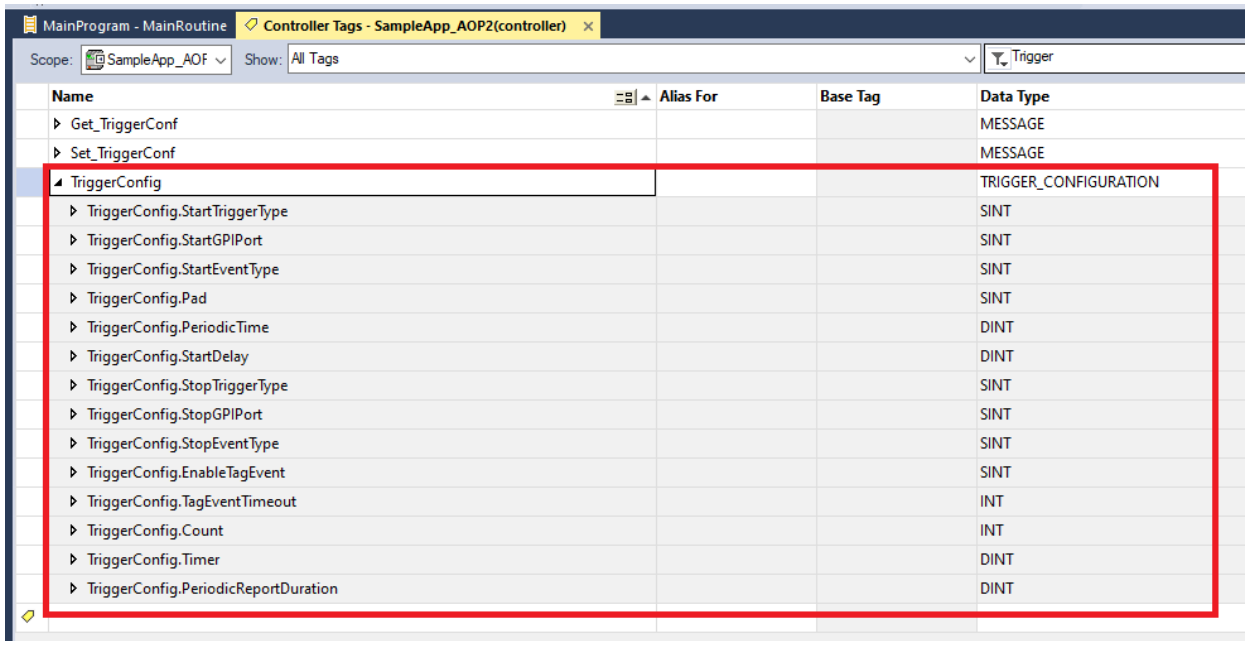


To get the value for configured triggers:

1. Enable Rung 11 by toggling ENABLE.GetTriggerConf Examine On Instruction. This reads the trigger configuration values in the TriggerConfig tag in controller tags. By default, there is no trigger configuration applied so all the values are 0.
2. Configure triggers as per the requirement by modifying the values in TriggerConfig. StartTriggerType and StopTriggerType fields specify the trigger type and corresponding values should be set to take effect. A detailed description is provided in the Zebra EtherNet/IP Data Model document.

NOTE: The Zebra EtherNet/IP Data Model document is included in the EtherNet/IP deliverable and is also available at: www.zebra.com/support.

Figure 66 Trigger Values

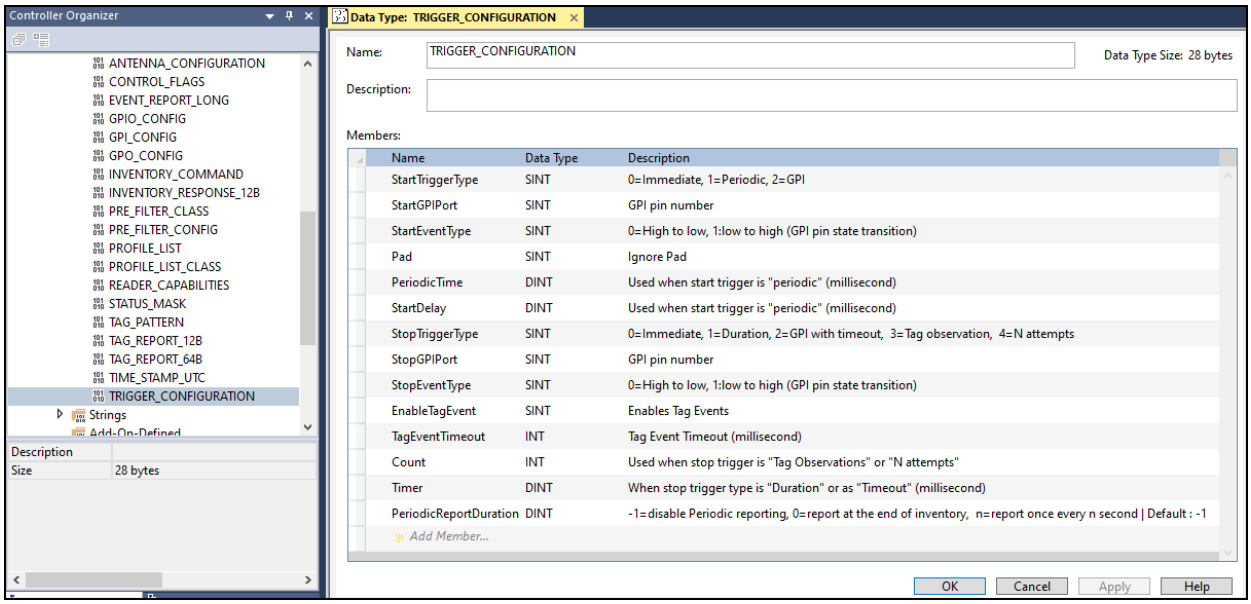


3. Select the **MainProgram - MainRoutine** tab and enable Rung 12 by toggling ENABLE.SetTriggerConf Examine On Instruction. This applies the trigger configuration in the reader.
4. To verify, perform GetTriggerConfig to validate the values.

Trigger Configuration Parameters:

- Vendor Class: 69

- Data type size: 28 bytes

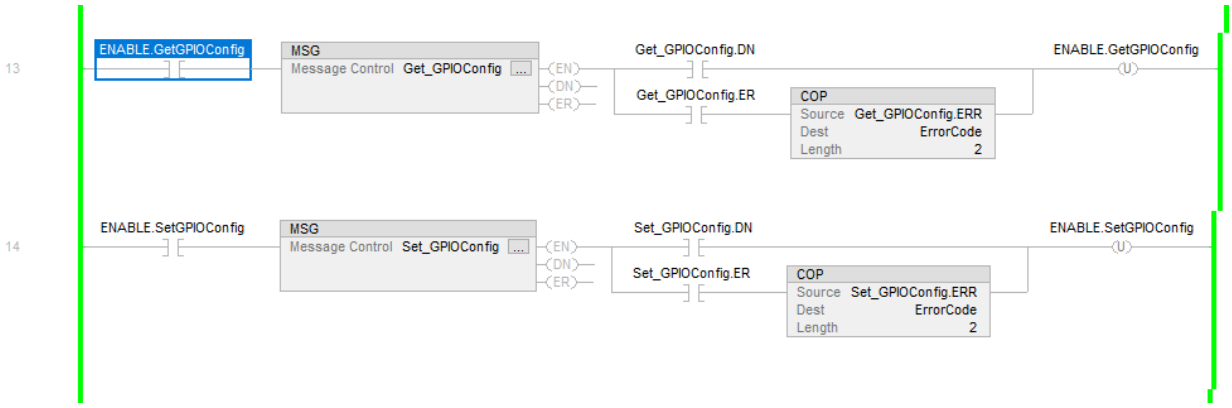


GPIO Configuration

The FX9600 RFID reader has a number of optically isolated GPIs and GPOs exported as a GPIO interface connection. Refer to the FX Series Integrator Guide for information about the GPIO interface in the FX9600 reader.

The status of GPI and GPO pins can be obtained with GPIO configuration explicit messaging available in the EIP model. Rung 13 and Rung 14 were created to demonstrate the use of GPIO configuration.

Figure 67 GPIO Configuration

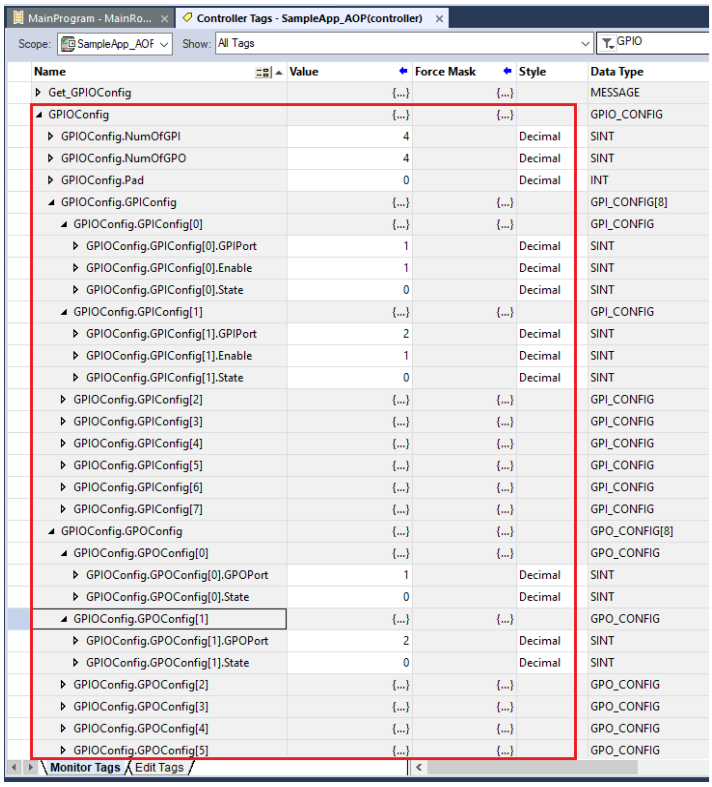


To get the current status of GPIOs:

1. Enable Rung 13 by toggling ENABLE.GetGPIOConfig Examine On Instruction. This reads the trigger configuration values in the GPIOConfig tag in controller tags.

   As shown in Figure 68, the NumOfGPI and NumOfGPO fields display the number of GPI and GPO available. In the config values array same number of elements have valid data.

Figure 68 GPIOConfig Tags



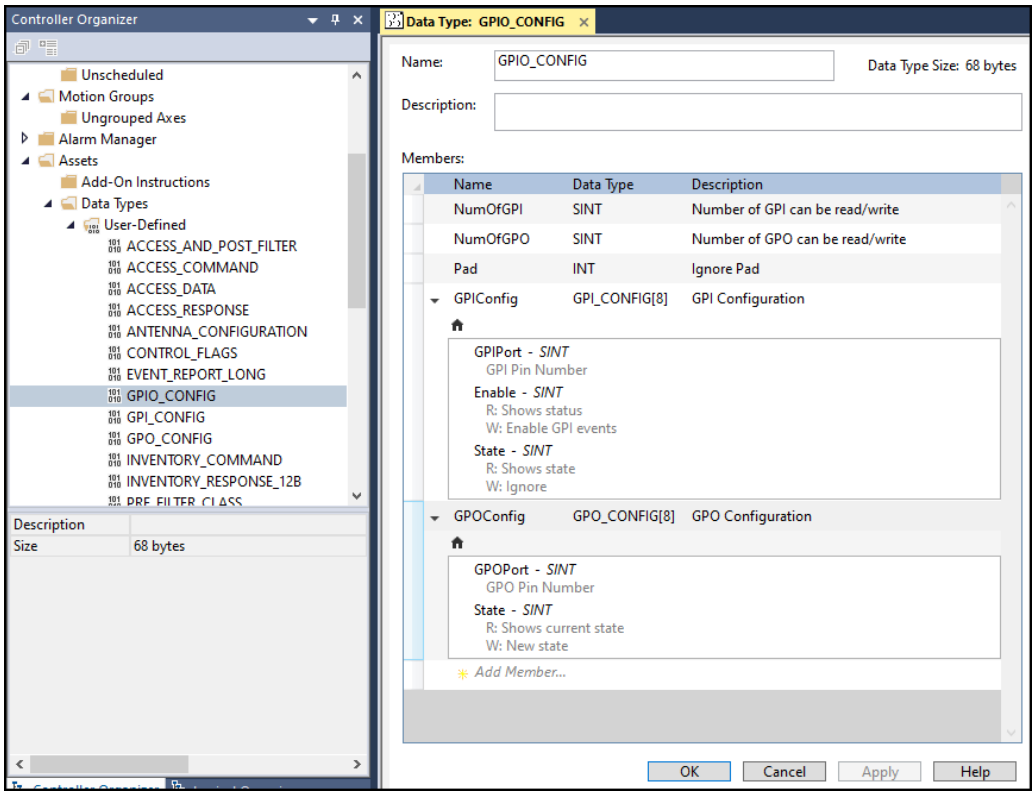
2. To enable the GPI or set the GPO status:
   a. Enter, in the respective NumOfGPI and NumOfGPO fields, the number of GPI and GPO values (i.e., how many GPIs and GPOs need to be updated.
   b. Fill that number of GPI and GPO config arrays with values to be updated.
3. After entering the values, select the MainProgram - MainRoutine tab and enable Rung 14 by toggling ENABLE.SetGPIOConfig Examine On Instruction. This applies the GPIO configuration in the reader.

NOTE: GPI status is a GET only parameter; GPI can be enabled/disabled only but cannot set the status.

GPIO Configuration Parameters:

- Vendor class: 6A

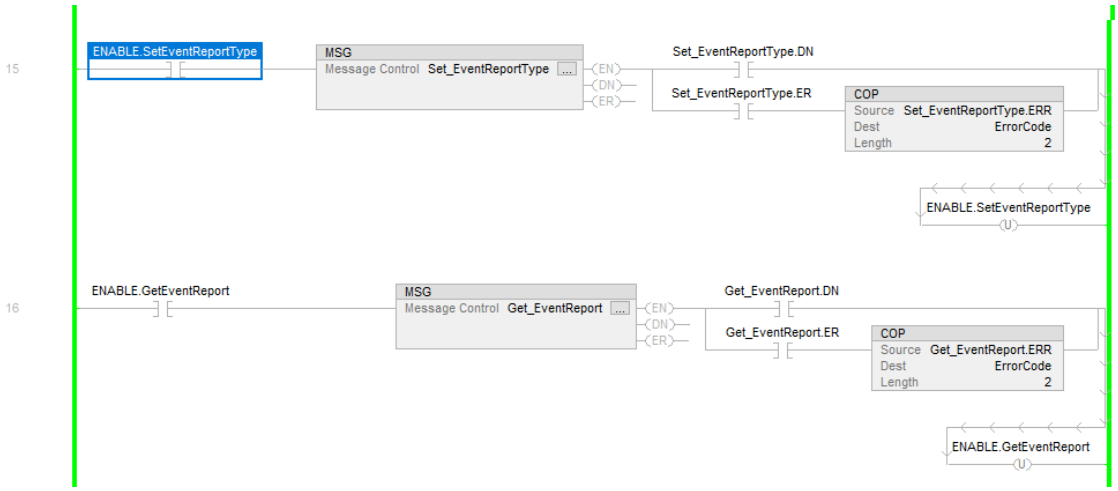
- Data type size: 68 bytes



Reading Event Report

The FX9600 RFID reader generates events which can be checked from the Inventory or Access response StatusMask.EventStatus field. Once an event is generated, the user can read the event report with explicit messaging. Rung 15 and Rung 16 were created to demonstrate the functionality.

Figure 69 Generate Events

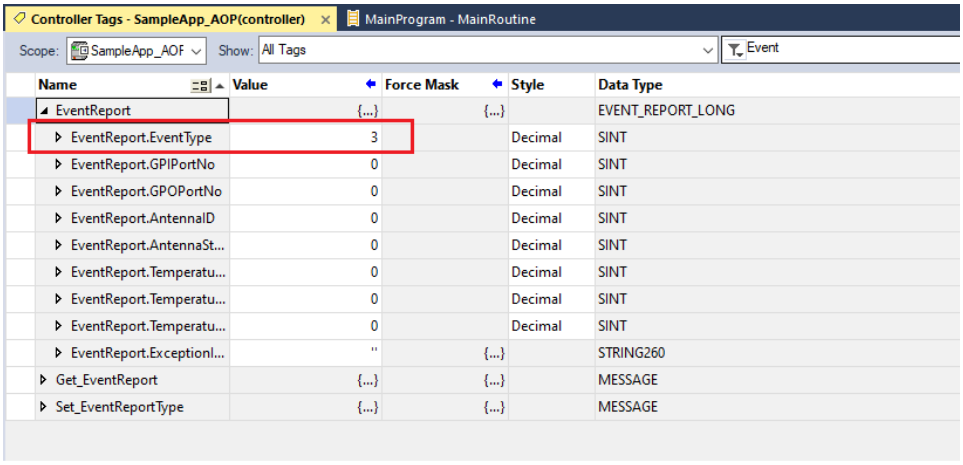


To get the event report:

1. Set the event report type for which event report to read by entering the EventType field in the EventReport tag.

2. Select the MainProgram - MainRoutine tab and enable Rung 15 by toggling ENABLE.SetEventReportType Examine On Instruction. This applies the Event Type in the reader.

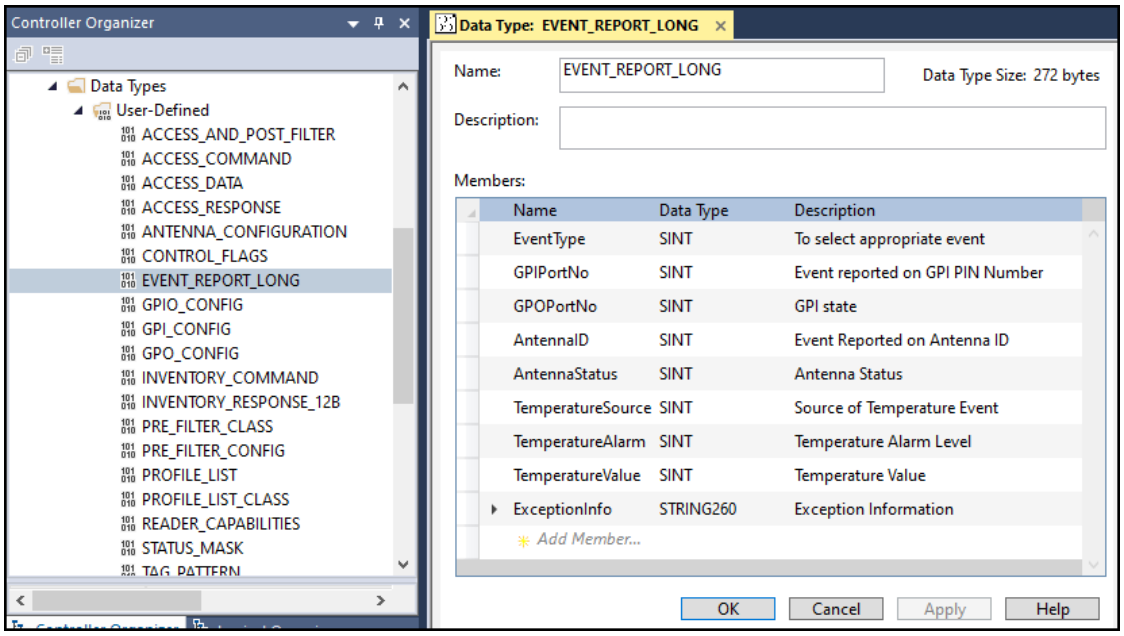
Figure 70 Set Event Type



3. To read the Event Report, enable Rung 16 by toggling ENABLE.GetEventReport Examine On Instruction. This reads the Event Report in event related fields in the EventReport tag.

Event Report Configuration Parameters:

- Vendor class: 6C
- Data type size: 272 bytes

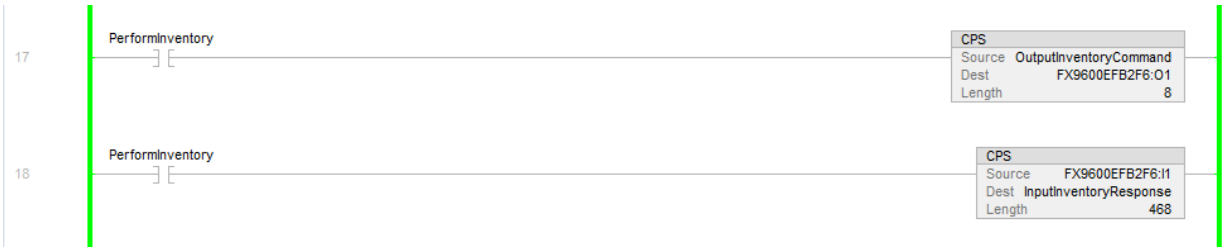


Performing Inventory

The Zebra RFID EtherNet/IP can be used to perform Inventory operations using the RFID reader. This section explains the use of the Inventory Command and Response to perform Inventory operations and read tag data from the RFID reader.

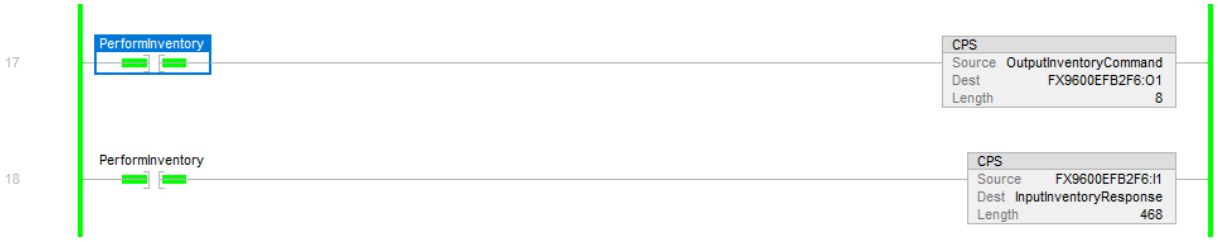
The sample application can perform the Inventory operation using two Rungs; Rung 17 to copy inventory command values from the local structure to the module defined structure, and Rung 18 to copy inventory response packet values from the module defined structure to the local structure.

Figure 71 Inventory Operation



1. Enable Rungs17 and Rung 18 by toggling the Rungs Examine On Instruction with the PerformInventory tag.
2. When enabled, the instruction turns green to show the enabled status.

Figure 72 Green Enabled Status



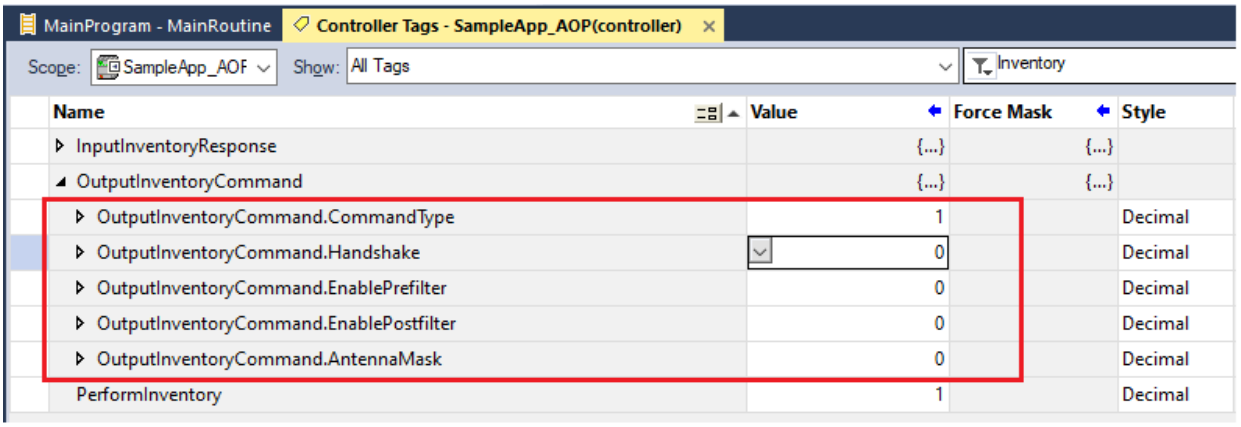
3. After enabling the run, perform the Inventory operation.

Start Inventory Operation

To start the Inventory operation with the reader:

1. Modify the OutputInventoryCommand tag structure.

Figure 73 Modifying OutputInventoryCommand



2. To start the Inventory operation, modify the **OutputInventoryCommand** fields as described in Table 3 and then increment Handshake by 1.
3. When the Handshake value is incremented, this command performs with the specified values and starts the Inventory operation.

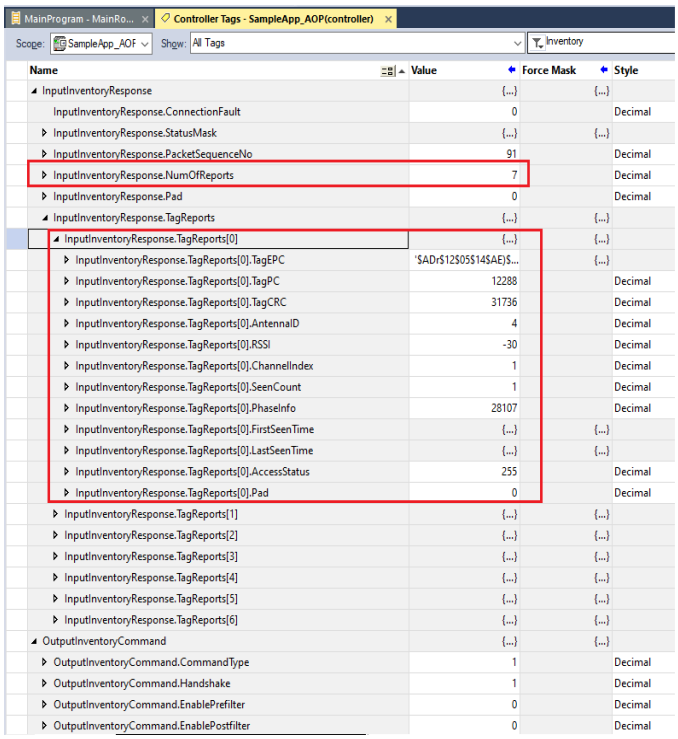
Table 3 OutputInventoryCommand Field Descriptions

Field	Description
Command Type	This field is used to specify the inventory command. It can be either 1 (START) or 2 (STOP).
Handshake	This should be modified at the end. This is the incremental handshake value to notify the reader that this is a new packet so the reader can perform the operation on the basis of this packet. Every new command packet should increment the Handshake field by 1. The Handshake value can be incremented to 127 (max). When the Handshake value reaches 127 it re-starts at zero.
EnablePrefilter	Used to specify whether or not the user wants to use pre-filter with this inventory command. Non-zero value is treated as TRUE and pre-filter takes effect.
EnablePostfilter	Used to specify whether or not post-filter is used. Criteria is the same as pre-filter.
AntennaMask	Use this mask to specify the antenna IDs on which this Inventory operation should perform. Each bit in the antenna mask, from 0 bit, represents one antenna. 0 bit for antenna ID 1, and so on. If the AntennaMask is 0, then inventory is performed with all available antennas.

Read Inventory Data

When the inventory starts, the sample application starts reading inventory data in InputInventoryResponse tag. TAG data is read as part of the InputInventoryResponse tag in TAGReports as array. Each element in the array is one TAG data. The number of TAG data in the array is specified in the NumOfReports field.

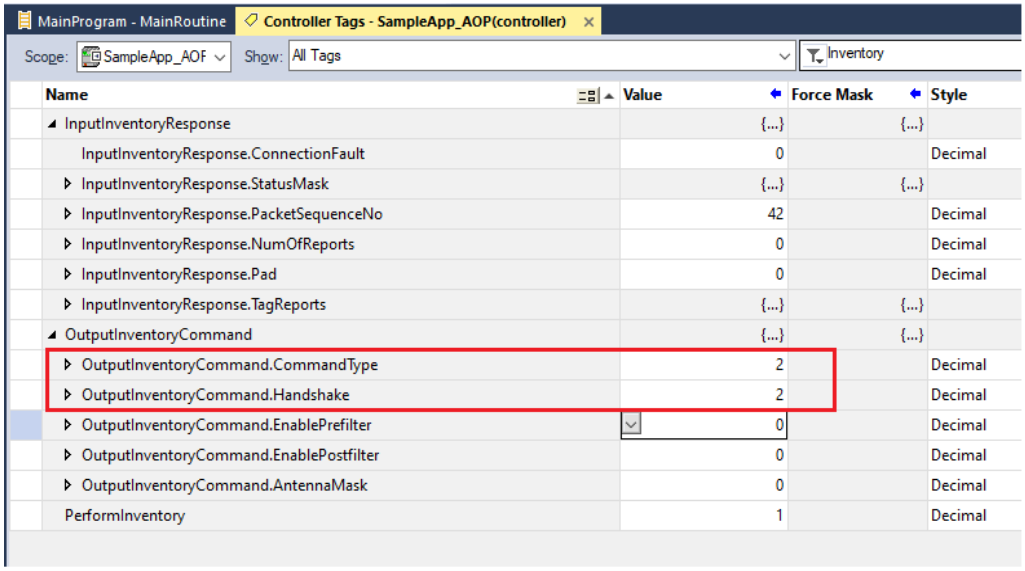
Figure 74 Reading Inventory Data



Stop Inventory Operation

The Inventory operation can be stopped by modifying CommandType as 2 (STOP) and by incrementing the Handshake value in the OutputInventoryCommand tag. The remaining fields are ignored if CommandType is 2 (STOP).

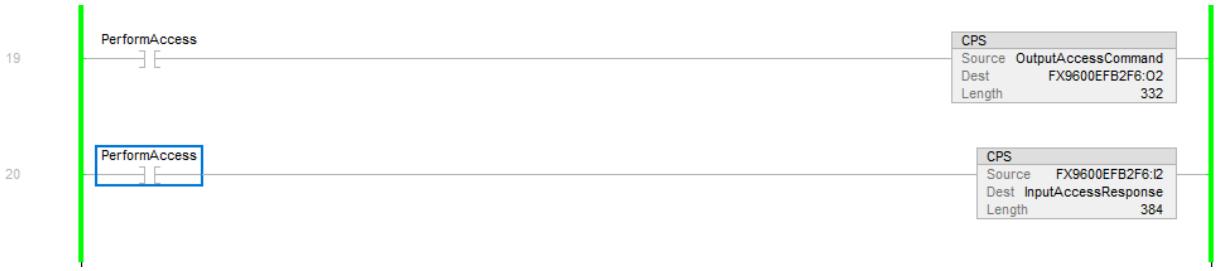
Figure 75 Stop Inventory Operation



Performing Access Operation

This section explains the use of the EIP sample application to perform the Access Operation and Reading the Access data. In the sample application Rung 19 and Rung 20 were created to perform the Access Operation with the reader using the EtherNet/IP protocol.

Figure 76 Access Operation/Read Access Data



To perform the Access operation with the reader:

1. Enable Rung 19 and Rung 20 by toggling the PerformAccess tag. When Rung 19 and Rung 20 are enabled, the Access Output Command data copies from the OutputAccessCommand to the internal Output assembly O2. Rung 20 also copies Access Response data from the input assembly I2 to the local InputAccessResponse tag.

Figure 77 Access Operation

2. When Rung 19 and Rung 20 are enabled, data in the command tag can be modified to perform the Access operation.

NOTE: Access and Inventory operations cannot be performed together. If Inventory is running it must be stopped before running the Access operation.

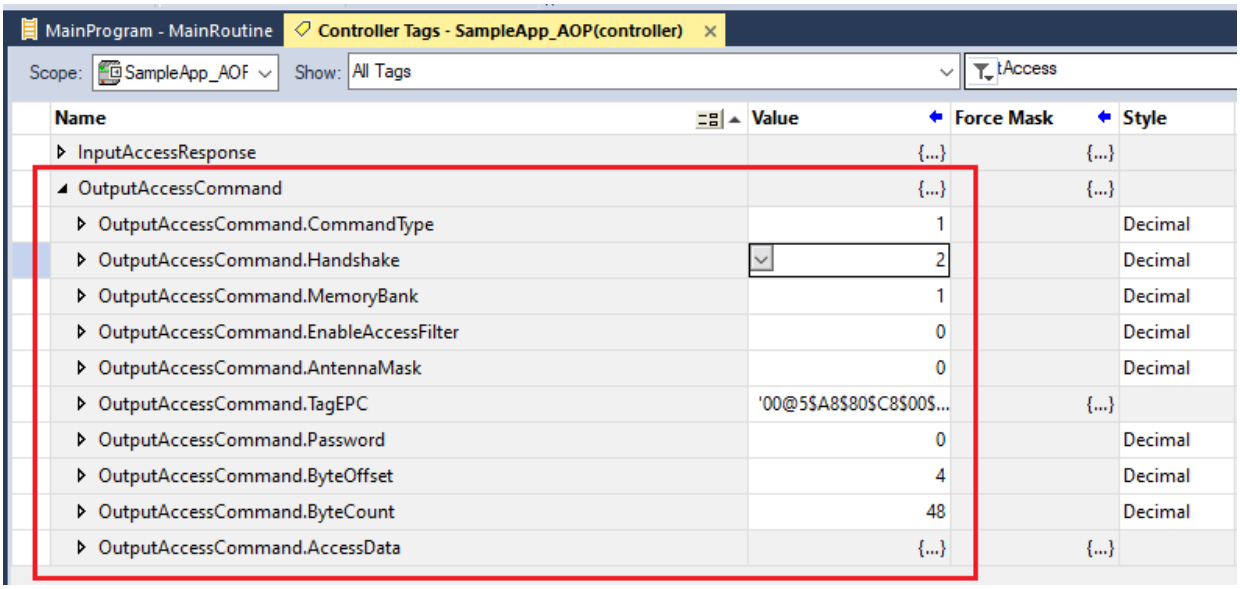
Access Read Operation

To perform the Access Read operation:

1. Specify the Command Type in OutputAccessCommand as 1 (ACCESS_READ).
2. Specify the following Read Access parameters (required to perform the Read operation).
   a. Enable Access Filter, if the Access filter needs to be applied
   b. Antenna Mask
   c. TAG EPC, if the Read operation needs to be performed on a specific EPC
   d. Password
   e. ByteOffset
   f. ByteCount

NOTE: AccessData is required when performing AccessWrite.

Figure 78 Access Read Operation



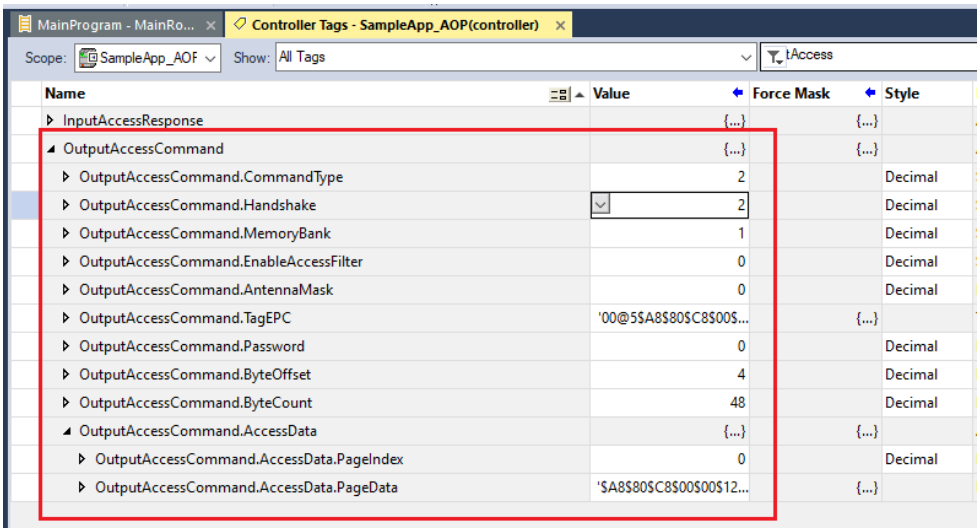
1. When all the parameters are set, increment the Handshake value. When the Handshake value is incremented, the EIP application running inside reader consumes the packet and performs the Access Read operation with the specified parameters.
2. The Access operation performs for one round, then stops. Access Data is reported in the InputAccessResponse tag.

Access Write Operation

The Access Write operation requires the same parameters as the Access Read Operation on page 69 with the exception of the following items

- Command Type is 2 (ACCESS_WRITE)
- AccessData must be filled with data to be written.

Figure 79 Access Write Operation

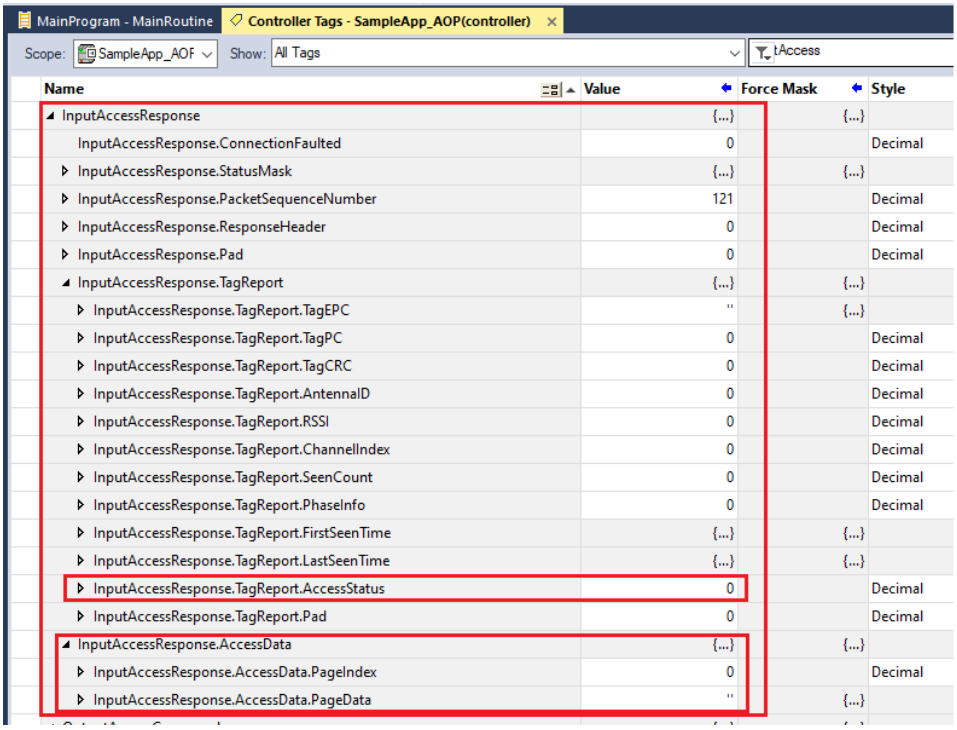


When all the parameters are sent, increment the Handshake value for the operation to be performed.

Access Response Data

The response from the Access Operation is received in the InputAccessResponse tag.

Figure 80 Access Response Data



Error Codes and Troubleshooting

Error Codes

There are two different categories of error codes reported in-case of the failure of performing any operation/command with FX9600 EtherNet/IP stack. These categories are defined below.

- EtherNet/IP Stack Error
- RFID Operation Specific Error.

EtherNet/IP Stack Error

Table 4 lists the error codes caused by a malfunction in the corresponding command/operation performed with the FX9600 EtherNet/IP stack. Usually the error is due to an incorrect size specification in the operation/command or invalid access of data member. These error codes are reported as the execution response of the command/operation itself and these are reported as per the standard EtherNet/IP operation in Studio 5000 Logix Designer.

Table 4 EtherNet/IP Stack Errors

Error Code (Hex)	Error
0x00	ERR_SUCCESS
0x01	ERR_CNXN_FAILURE
0x02	ERR_RESOURCE_UNAVAIL
0x03	ERR_INV_PARAMNAME
0x04	ERR_PATHSEGMENT
0x05	ERR_PATHDESTUNKNOWN
0x06	ERR_PARTIALXFER
0x07	ERR_CNXNLOST
0x08	ERR_SERV_UNSUPP
0x09	ERR_INV_ATTRIBVAL
0x0A	ERR_ATTR_LIST_ERR
0x0B	ERR_IN_REQ_STATE
0x0C	ERR_OBJ_STATE_CONFLICT
0x0D	ERR_OBJ_ALREADY_EXISTS
0x0E	ERR_ATTR_READONLY

Table 4 EtherNet/IP Stack Errors (Continued)

Error Code (Hex)	Error
0x0F	ERR_PRIV_VIOLATION
0x10	ERR_DEV_STATE_CONFLICT
0x11	ERR_REPLY_SIZE
0x12	ERR_FRAG_PRIM_VAL
0x13	ERR_INSUFF_DATA
0x14	ERR_ATTR_UNSUPP
0x15	ERR_TOOMUCH_DATA
0x16	ERR_UNEXISTANT_OBJ
0x17	ERR_SERV_FRAG_SEQ
0x18	ERR_NO_ATTR_DATA
0x19	ERR_STORE_FAILURE
0x1A	ERR_ROUTE_REQ_TOO_LRG
0x1B	ERR_ROUTE_RSP_TOO_LRG
0x1C	ERR_MISSING_ATTR_LIST
0x1D	ERR_INV_ATTR_LIST
0x1E	ERR_EMBEDDED_SERV_ERR
0x1F	ERR_VENDOR_SPECIFIC
0x20	ERR_INV_SERVICE_PARM
0x21	ERR_WO_VAL_ALREADY_W
0x22	ERR_INV_REP_RECV
0x23	ERR_BUFFER_OVERFLOW
0x24	ERR_MSG_FORMAT_ERR
0x25	ERR_KEY_ERR_IN_PATH
0x26	ERR_PATH_SIZE_INV
0x27	ERR_UNEXP_ATTR_IN_LIST
0x28	ERR_INV_MEMBER_ID
0x29	ERR_MEMBER_READONLY
0x2A	ERR_G2ONLY_GEN_ERR
0x2B	ERR_UNKNOWN_MB_ERR
0x2C	ERR_ATTR_NOT_GETTABLE

RFID Operation Specific Error

Table 5 lists the error codes while performing an RFID specific operation. These errors occur from invalid/incorrect values of any configuration parameter or any failure in the execution of the command/operation at the RFID layer. These error codes are reported as part of the response data for Inventory and Access operations in the Error Status field. This is the sticky error code and it is cleared after the next successful assembly operation (Inventory or Access operation) only.

Table 5 RFID Operation Specific Errors

Error Code (Decimal)	Error Code (Text)
0	RFID_API_SUCCESS
1	RFID_API_COMMAND_TIMEOUT
2	RFID_API_PARAM_ERROR
3	RFID_API_PARAM_OUT_OF_RANGE
4	RFID_API_CANNOT_ALLOC_MEM
5	RFID_API_UNKNOWN_ERROR
6	RFID_API_INVALID_HANDLE
7	RFID_API_BUFFER_TOO_SMALL
8	RFID_READER_FUNCTION_UNSUPPORTED
9	RFID_RECONNECT_FAILED
10	RFID_API_DATA_NOT_INITIALISED
11	RFID_API_ZONE_ID_ALREADY_EXITS
12	RFID_API_ZONE_ID_NOT_FOUND
100	RFID_COMM_OPEN_ERROR
101	RFID_COMM_CONNECTION_ALREADY_EXISTS
102	RFID_COMM_RESOLVE_ERROR
103	RFID_COMM_SEND_ERROR
104	RFID_COMM_RECV_ERROR
105	RFID_COMM_NO_CONNECTION
106	RFID_INVALID_SOCKET
107	RFID_READER_REGION_NOT_CONFIGURED
108	RFID_READER_REINITIALIZING
109	RFID_SECURE_CONNECTION_ERROR
110	RFID_ROOT_SECURITY_CERTIFICATE_ERROR
111	RFID_HOST_SECURITY_CERTIFICATE_ERROR
112	RFID_HOST_SECURITY_KEY_ERROR
200	RFID_CONFIG_GET_FAILED
201	RFID_CONFIG_SET_FAILED
202	RFID_CONFIG_NOT_SUPPORTED
300	RFID_CAP_NOT_SUPPORTED

Table 5 RFID Operation Specific Errors (Continued)

Error Code (Decimal)	Error Code (Text)
301	RFID_CAP_GET_FAILED
400	RFID_FILTER_NO_FILTER
401	RFID_FILTER_INVALID_INDEX
402	RFID_FILTER_MAX_FILTERS_EXCEEDED
403	RFID_NO_READ_TAGS
404	RFID_NO_REPORTED_EVENTS
405	RFID_INVENTORY_MAX_TAGS_EXCEEDED
406	RFID_INVENTORY_IN_PROGRESS
407	RFID_NO_INVENTORY_IN_PROGRESS
420	RFID_TAG_LOCATING_IN_PROGRESS
421	RFID_NO_TAG_LOCATING_IN_PROGRESS
422	RFID_NXP_EAS_SCAN_IN_PROGRESS
423	RFID_NO_NXP_EAS_SCAN_IN_PROGRESS
500	RFID_ACCESS_IN_PROGRESS
501	RFID_NO_ACCESS_IN_PROGRESS
502	RFID_ACCESS_TAG_READ_FAILED
503	RFID_ACCESS_TAG_WRITE_FAILED
504	RFID_ACCESS_TAG_LOCK_FAILED
505	RFID_ACCESS_TAG_KILL_FAILED
506	RFID_ACCESS_TAG_BLOCK_ERASE_FAILED
507	RFID_ACCESS_TAG_BLOCK_WRITE_FAILED
508	RFID_ACCESS_TAG_NOT_FOUND
510	RFID_ACCESS_SEQUENCE_NOT_INITIALIZED
511	RFID_ACCESS_SEQUENCE_EMPTY
512	RFID_ACCESS_SEQUENCE_IN_USE
513	RFID_ACCESS_SEQUENCE_MAX_OP_EXCEEDED
514	RFID_ACCESS_TAG_RECOMMISSION_FAILED
515	RFID_ACCESS_TAG_BLOCK_PERMALOCK_FAILED
516	RFID_ACCESS_NXP_TAG_SET_EAS_FAILED
517	RFID_ACCESS_NXP_TAG_READ_PROTECT_FAILED
518	RFID_ACCESS_FUJITSU_CHANGE_WORDLOCK_FAILED
519	RFID_ACCESS_FUJITSU_CHANGE_BLOCKLOCK_FAILED
520	RFID_ACCESS_FUJITSU_READ_BLOCKLOCK_FAILED
521	RFID_ACCESS_FUJITSU_BURST_WRITE_FAILED
522	RFID_ACCESS_FUJITSU_BURST_ERASE_FAILED

Table 5 RFID Operation Specific Errors (Continued)

Error Code (Decimal)	Error Code (Text)
523	RFID_ACCESS_FUJITSU_CHANGE_BLOCK_OR_AREA_GROUPPASSWORD_FAILED
524	RFID_ACCESS_FUJITSU_AREA_READLOCK_FAILED
525	RFID_ACCESS_FUJITSU_AREA_WRITELOCK_FAILED
526	RFID_ACCESS_FUJITSU_AREA_WRITELOCK_WOPASSWORD_FAILED
527	RFID_ACCESS_NXP_CHANGE_CONFIG_FAILED
528	RFID_ACCESS_IMPINJ_QT_READ_FAILED
529	RFID_ACCESS_IMPINJ_QT_WRITE_FAILED
530	RFID_ACCESS_G2V2_AUTHENTICATE_FAILED
531	RFID_ACCESS_G2V2_READBUFFER_FAILED
532	RFID_ACCESS_G2V2_UNTRACEABLE_FAILED
533	RFID_ACCESS_G2V2_CRYPTO_FAILED
601	RFID_RM_INVALID_USERNAME_PASSWORD
602	RFID_RM_NO_UPDATION_IN_PROGRESS
603	RFID_RM_UPDATION_IN_PROGRESS
604	RFID_RM_COMMAND_FAILED
605	RFID_NXP_BRANDID_CHECK_IN_PROGRESS
606	RFID_NO_RF_SURVEY_OPERATION_IN_PROGRESS
607	RFID_RFSURVEY_IN_PROGRESS
700	RFID_INVALID_ERROR_CODE

Troubleshooting

Use the following troubleshooting information to solve some common problems faced in setting up a sample application with Studio 5000 Logix Designer.

Unable to Load Sample Application

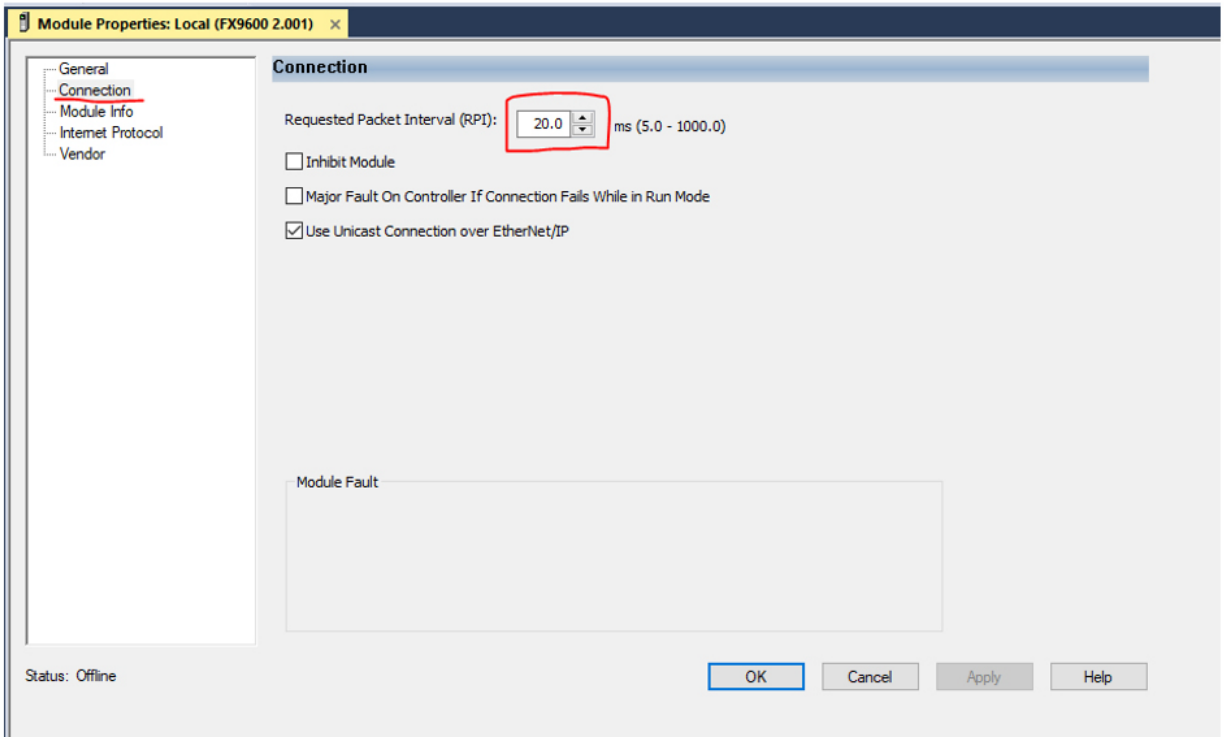
If the sample application is not able to load directly, this can be because of an incompatible version of Studio 5000 Logix Designer or PLC series. In this case, the user must create a new project for the PLC and load the Rungs/Data Types as mentioned in Using Non-compatible Versions of Studio 5000 Logix Designer and PLC on page 17.

Setting Proper Requested Packet Interval (RPI)

There can be a situation in which Studio 5000 Logix Designer is not able to process data received from the reader because the default RPI is set to 10 ms. In this scenario, the user must configure the project with the proper RPI to suit user requirements.

Figure 81 shows the RPI configuration window.

Figure 81 Module Properties Window



I/O Not Responding Status

Upon running the program from Studio 5000 Logix Designer, when status shows that the I/O is not responding, there can be a few scenarios causing this issue.

Application EtherNet/IP Adapter Application Not Running

Studio 5000 Logix Designer reports an I/O not responding error if the application is not running in the reader. If the application is not running, PLC cannot connect because the EtherNet/IP port is not opened in reader.

In this case, the user must check the application status from the reader web-console. If it is not running, start the application by clicking on the **Start** button.

Reader IP Not Configured/Reachable

Another scenario for the I/O not responding can occur because PLC is not able to reach the reader. This is possible when the reader and PLC are in different subnets and the IP address is not configured properly.

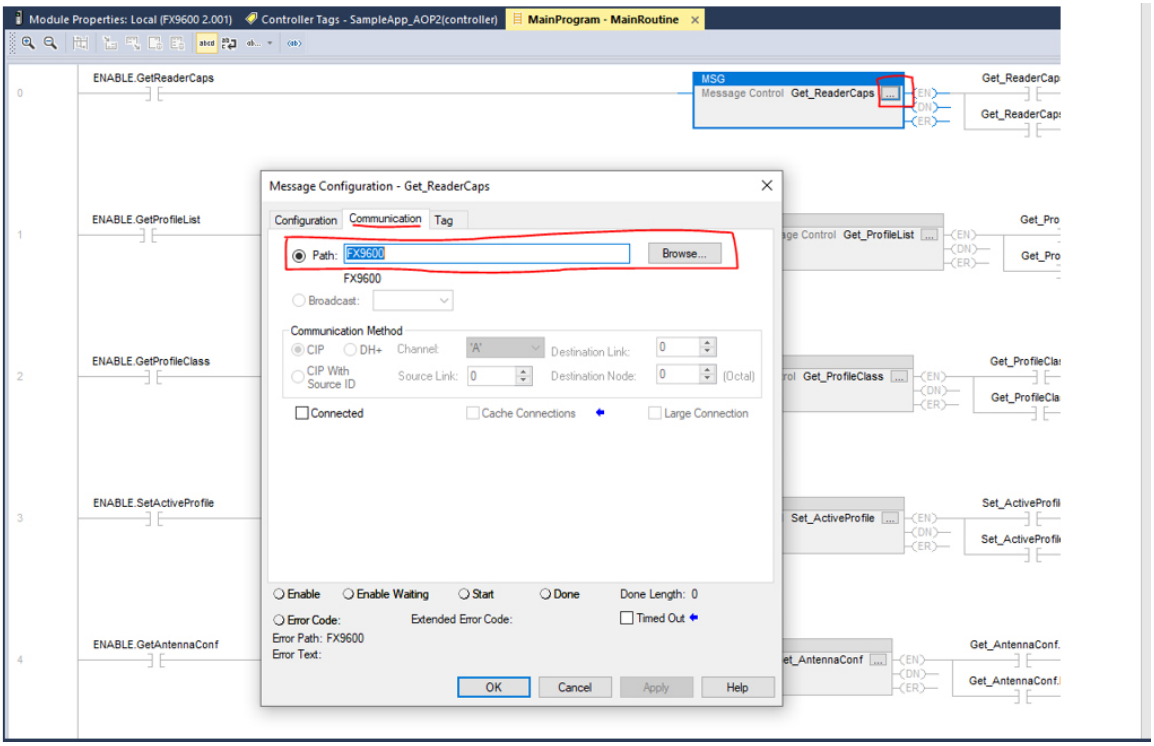
It is suggested that the reader and PLC remain in same subnet or assign the static IP address to both in the same subnet.

Communication Path Not Set Properly In MSG Instructions

If I/O is not responding and an error appears while performing explicit messaging, it could be because the communication path is not properly set for that specific explicit MSG instruction.

See Figure 82 to configure the correct communication path in each MSG instruction by editing the MSG instruction properties.

Figure 82 Module Properties Window



Unable to Perform Inventory or Access Operation

On performing the Inventory or Access operation, if the application is returning with an Unable to perform error, it may be possible that application is already running the previous command. As the Inventory and Access operations cannot be performed simultaneously, the user must stop the Inventory operation before running the Access operation, or wait for the Access operation to complete before running Inventory (if already running).

Response Data format

If data format is not showing in the expected format, it is required to change the data style in Studio 5000 Logix Designer to the required format. This can be done by selecting the proper style from drop-down menu for data, shown in Figure 83.

Figure 83 Style Window