

Allen-Bradley 5032 IO-Link Master Add-On Instruction Guide, v4 January 31st, 2025

This document covers the installation and use of Add-On Instruction (AOI) for the Logix Designer software package from Rockwell Automation. This AOI handles acyclic IO-Link commands from an Allen-Bradley PLC through an Allen-Bradley IO-Link Master. This AOI has three User Defined Tag data types.

This IO-Link Master AOI is meant to be used in conjunction with one or more v4 Banner IO-Link Device Parameter Data AOIs.

This document was written using Allen-Bradley Point I/O IO-Link Master (5032-8IOL). This **does not** work with the 1732 and 1734 IO-Link Masters.

Components

Banner_IOLM_A2_v4_AOI.L5X

UDT's Packaged with the AOI

Banner_IOLM_v4

Banner_IOLM_EL_v4

Banner_IOL_Port_v4

NOTE:

This Banner IO-Link Master AOI is useless on its own.

It is intended to be linked to one or more v4 Banner Device Parameter AOIs to function.

Other AOIs Available Separately

Banner has AOI files for other brands of IO-Link Master and for controlling a variety of Banner IO-Link devices. Banner also has AOI files for easily handling Banner device Process Data.

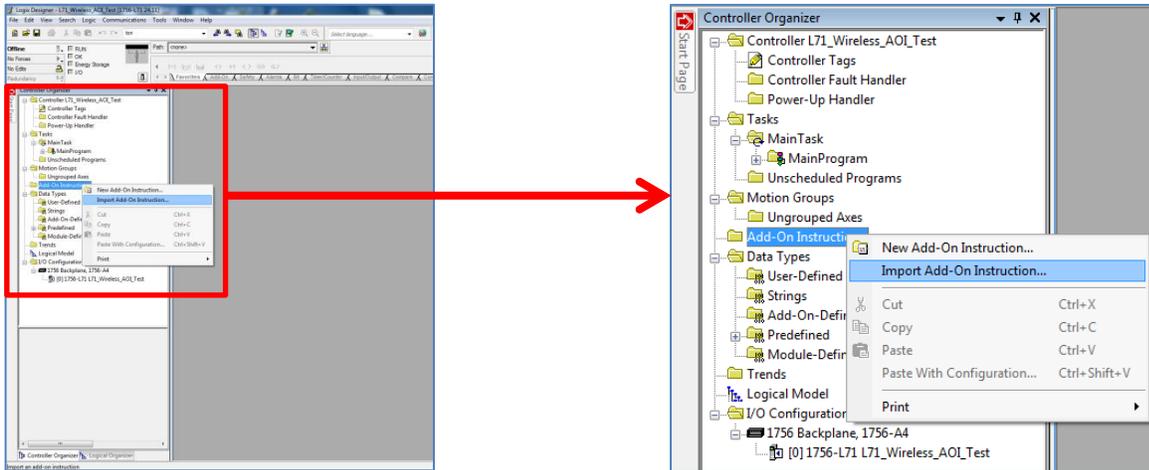
Contents

1.	Installation Process	1
2.	Configuring the Allen-Bradley IO-Link Master	3
3.	Configuring the Banner IO-Link Master AOI	4
4.	Linking the Master AOI to Device AOIs	12
5.	Using the Paired IO-Link Master and Device Parameter Data AOIs.....	13
Appendix A	IODD Device Integration	14
Appendix B	Error Handling & AOI Resets.....	15
Appendix C	Halt AOI Operation.....	16

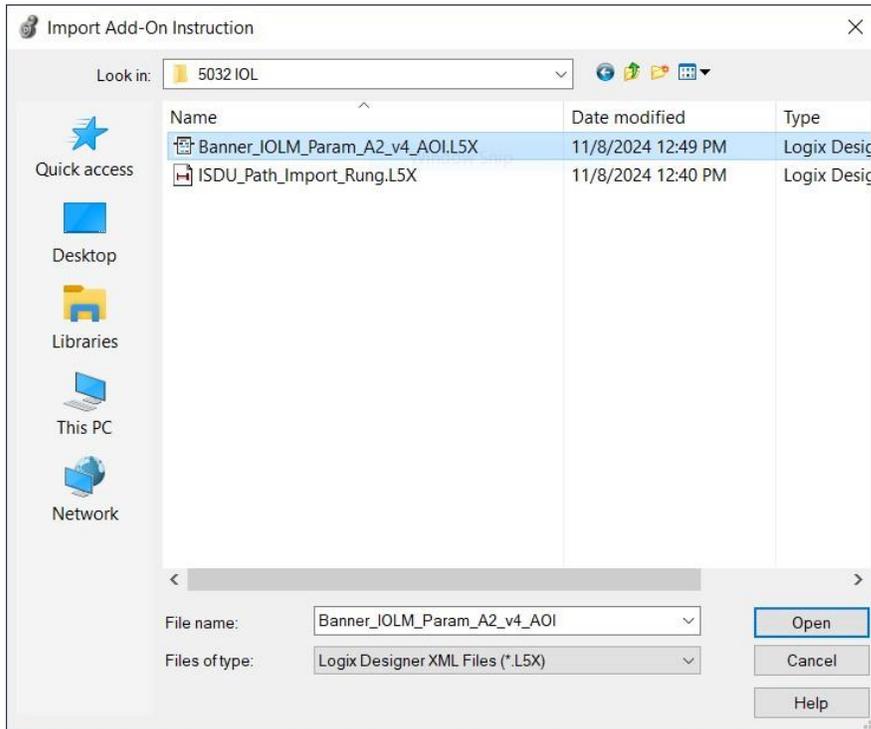
1. Installation Process

This section describes how to install the AOI in Logix Designer software.

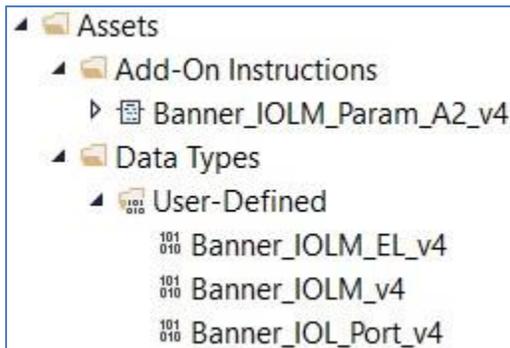
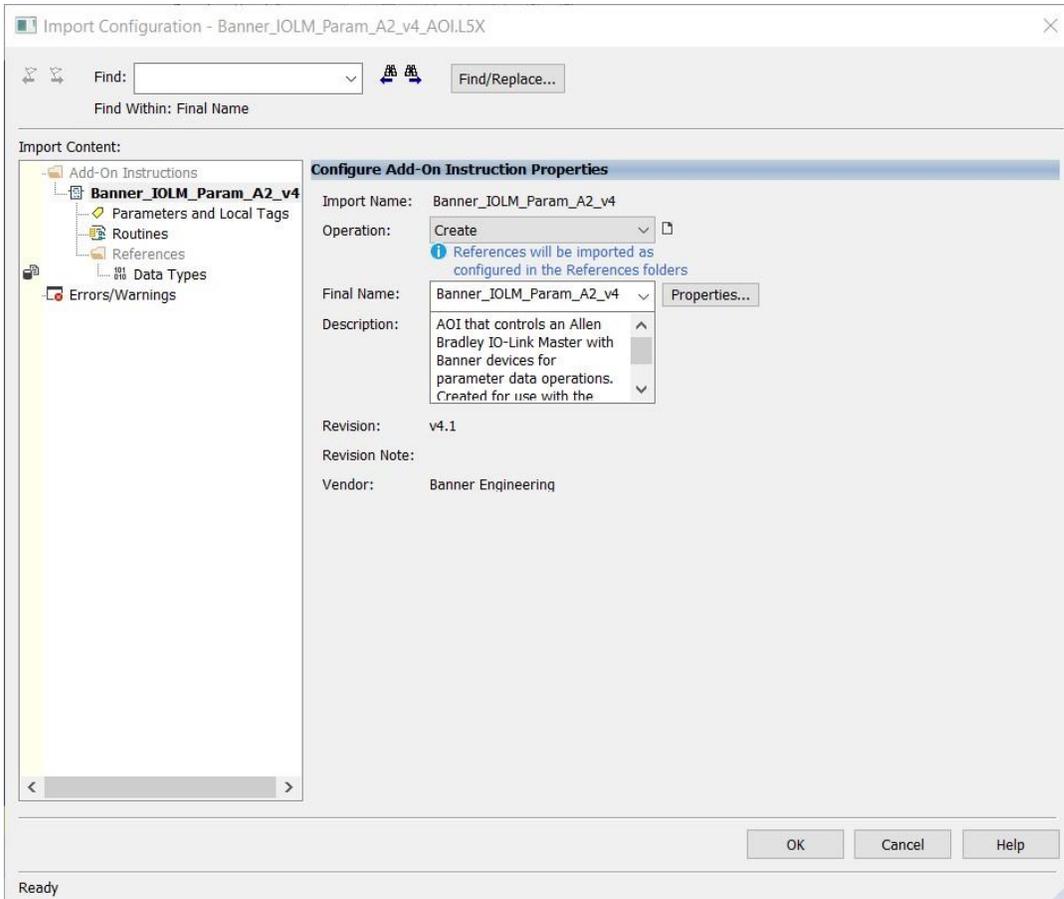
1. Open a project.
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.



3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner_IOLM_A2_v4_AOI.L5X" file will be selected. Click the Open button.



4. The Import Configuration window will pop up. The default selection will create all the necessary items for the AOI. Click the OK button to complete the import process.



5. The AOI is added to the Controller Organizer window and should look like the picture at left.
6. AOI installation into the Logix Designer software complete.

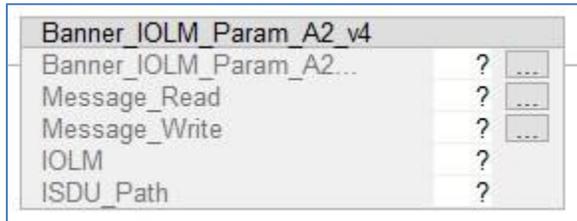
2. Configuring the Allen-Bradley IO-Link Master

Make an EtherNet/IP connection to the Allen-Bradley IO-Link Master.

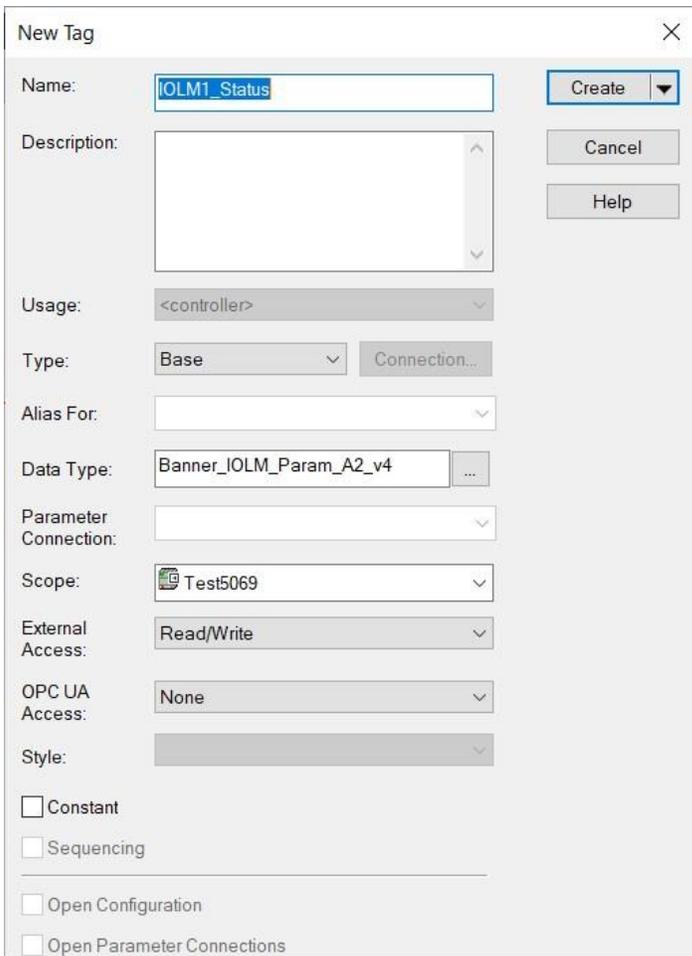
1. Create an Ethernet communications module for the Allen-Bradley IO-Link Master device. In this example the EDS file was used, and the connection was named "IOLM1".
2. Open the 5032 just created. Press the Change button. Set the ports that require IO-Link to IO-Link. Only even channels have IO-Link Capability. Press OK to save the changes.
3. Right click on the IO-Link icon under the 5032. Select New Module. Search for the IO-Link Device to add. If the IODD has yet to added See Appendix A of this document instructions on adding an IODD to Studio 5000.
4. Repeat the above steps as needed to add all IO-Link devices to the 5032 block.

3. Configuring the Banner IO-Link Master AOI

1. Add the “Banner_IOLM_A2_v4” AOI to your ladder logic program. For each of the question marks shown in the instruction we need to create and link a new tag. The AOI includes a new type of User Defined Tag (UDT): a custom array of tags meant specifically for this AOI.



2. In the AOI, right-click on the question mark on the line labeled “Banner_IOLM_A2_v4”. Click New Tag. In this example, we’ll use the name “IOLM1_Status”. The example naming convention accounts for this being the #1 IO-Link Master in our program. More masters could be named IOLM2, IOLM3, etc.



3. Now we set up the Messages used to read and write to devices connected to this IO-Link Master. Right click on the question mark for the "Message_Read" line in the AOI and choose New Tag. In this example we'll use the tag name "IOLM1_Read". Click Create.

Do the same for the "Message_Write" line in the AOI. In this example "IOLM1_Write" is used.

The screenshot shows the 'New Tag' dialog box with the following fields and values:

- Name: IOLM1_Read
- Description: (empty text area)
- Usage: <controller>
- Type: Base
- Alias For: (empty dropdown)
- Data Type: MESSAGE
- Parameter Connection: (empty dropdown)
- Scope: Test5069
- External Access: Read/Write
- OPC UA Access: None
- Style: (empty dropdown)
- Constant:
- Sequencing:
- Open MESSAGE Configuration:
- Open Parameter Connections:

The screenshot shows the 'New Tag' dialog box with the following fields and values:

- Name: IOLM1_Write
- Description: (empty text area)
- Usage: <controller>
- Type: Base
- Alias For: (empty dropdown)
- Data Type: MESSAGE
- Parameter Connection: (empty dropdown)
- Scope: Test5069
- External Access: Read/Write
- OPC UA Access: None
- Style: (empty dropdown)
- Constant:
- Sequencing:
- Open MESSAGE Configuration:
- Open Parameter Connections:

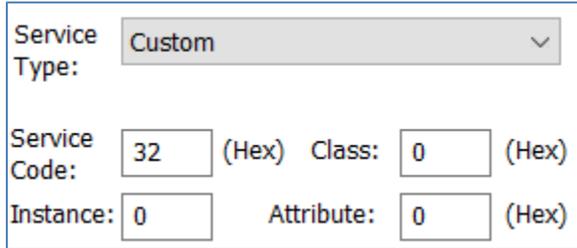
4. Now create a new tag array for the "IOLink" line in the AOI. Here we used the name "IOLM1". The tags created here will serve as linkages between the IO-Link Master AOI and the connected Banner device AOI(s). This group of tags also controls the flow of information to and from the master, ensuring that all sensors get a chance to read and write in an orderly fashion.

The screenshot shows the 'New Tag' dialog box with the following configuration:

- Name: IOLM1
- Description: (empty)
- Usage: <controller>
- Type: Base
- Alias For: (empty)
- Data Type: Banner_IOLM_v4
- Parameter Connection: (empty)
- Scope: Test5069
- External Access: Read/Write
- OPC UA Access: None
- Style: (empty)
- Constant:
- Sequencing:
- Open Configuration:
- Open Parameter Connections:

5. Create a new tag for the "ISDU_Path" line. This creates an array which stores the path for the controller to take to access the ISDU operation for the 5032 IO-Link Master.

- Now configure “Message_Read”, setting up the Explicit Message that will handle half of the communications between the PLC and the IO-Link Master. Click on the “...” button at the far right of the “Message_Read” line.
- In the Message Configuration window, keep the Message Type “CIP Generic” and the Service Type “Custom”. Enter a Service Code of 32.

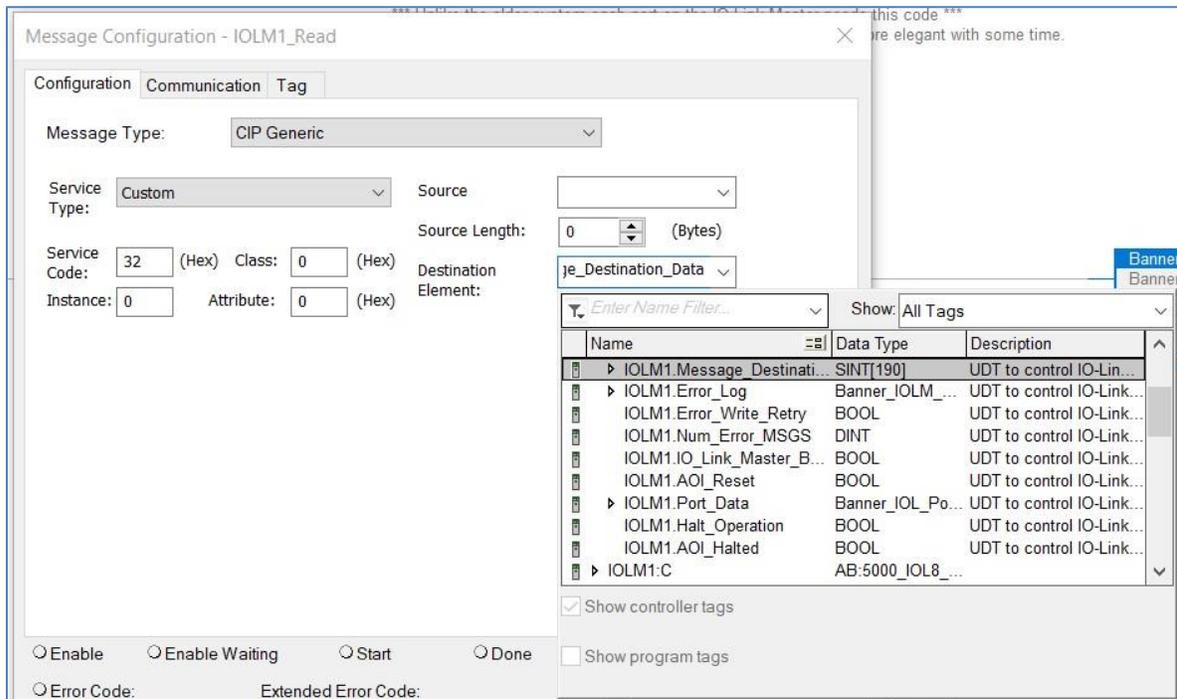


Service Type: Custom

Service Code: 32 (Hex) Class: 0 (Hex)

Instance: 0 Attribute: 0 (Hex)

- Set the Source Length to 0.
- For Destination Element, select “IOLM1.Message_Destination_Data”.



Message Configuration - IOLM1_Read

Configuration Communication Tag

Message Type: CIP Generic

Service Type: Custom Source: []

Service Code: 32 (Hex) Class: 0 (Hex) Source Length: 0 (Bytes)

Instance: 0 Attribute: 0 (Hex) Destination Element: IOLM1.Message_Destination_Data

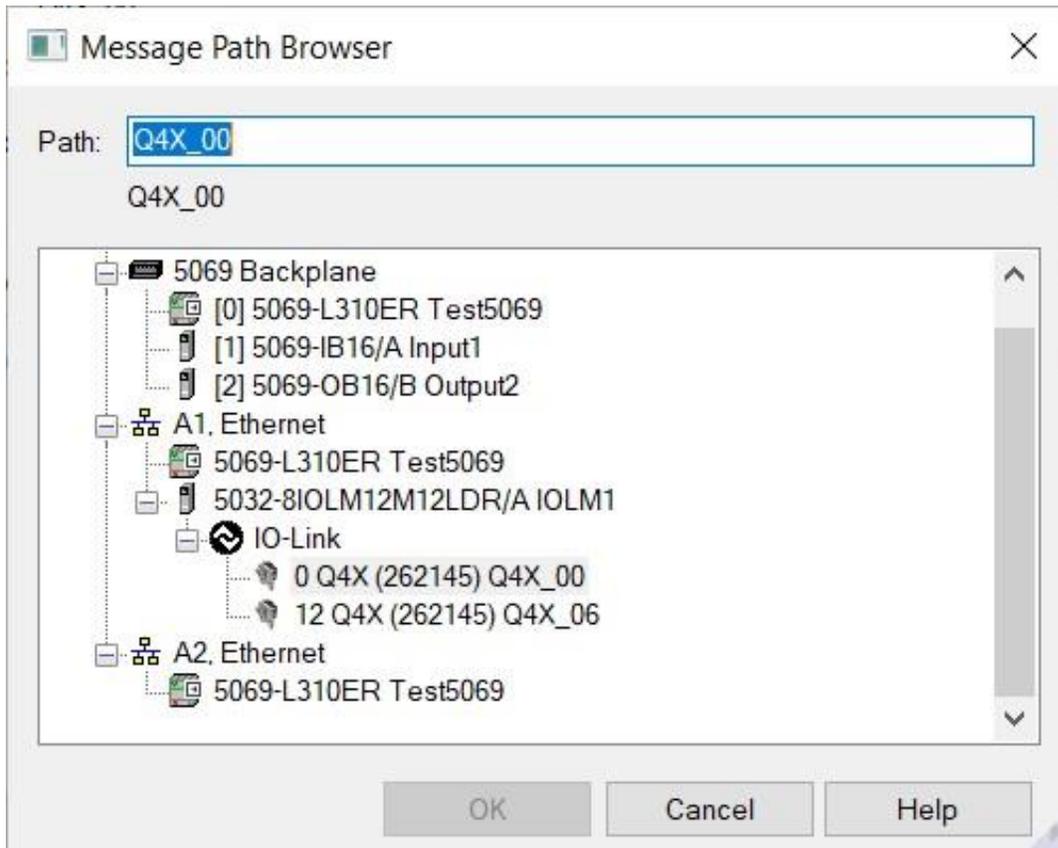
Name	Data Type	Description
IOLM1.Message_Destination_Data	SINT[190]	UDT to control IO-Link...
IOLM1.Error_Log	Banner_IOLM_...	UDT to control IO-Link...
IOLM1.Error_Write_Retry	BOOL	UDT to control IO-Link...
IOLM1.Num_Error_MSGS	DINT	UDT to control IO-Link...
IOLM1.IO_Link_Master_B...	BOOL	UDT to control IO-Link...
IOLM1.AOI_Reset	BOOL	UDT to control IO-Link...
IOLM1.Port_Data	Banner_IOL_Po...	UDT to control IO-Link...
IOLM1.Halt_Operation	BOOL	UDT to control IO-Link...
IOLM1.AOI_Halted	BOOL	UDT to control IO-Link...
IOLM1:C	AB:5000_IOL8_...	

Show controller tags Show program tags

Enable Enable Waiting Start Done

Error Code: Extended Error Code:

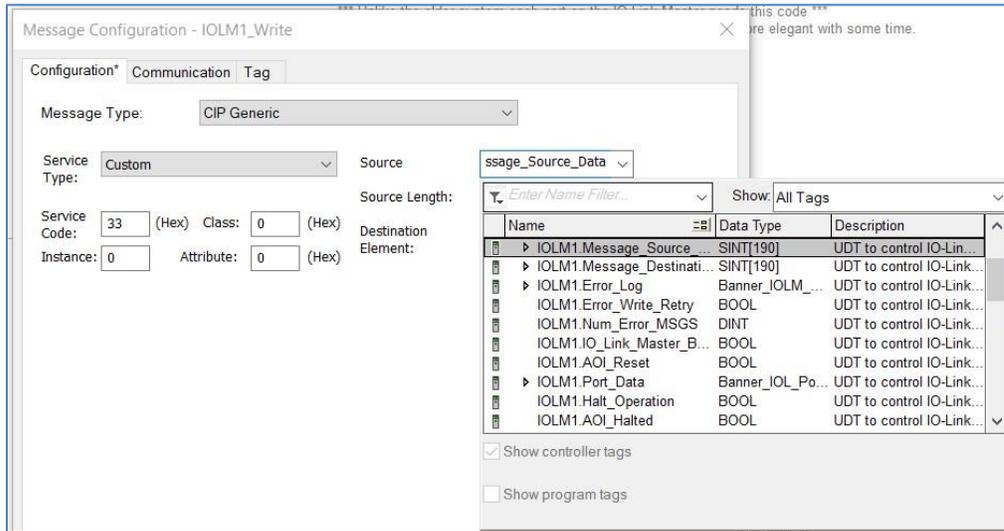
10. Now click on the Communication tab, then click the Browse button.
11. Select the IO-Link Master and then one of the IO-Link Devices attached to the unit, then click OK again to close the Message Configuration window.



12. Now configure “Message_Write”, setting up the Explicit Message that will handle the other half of the communications between the PLC and the IO-Link Master. Click on the “...” button at the far right of the “Message_Write” line.

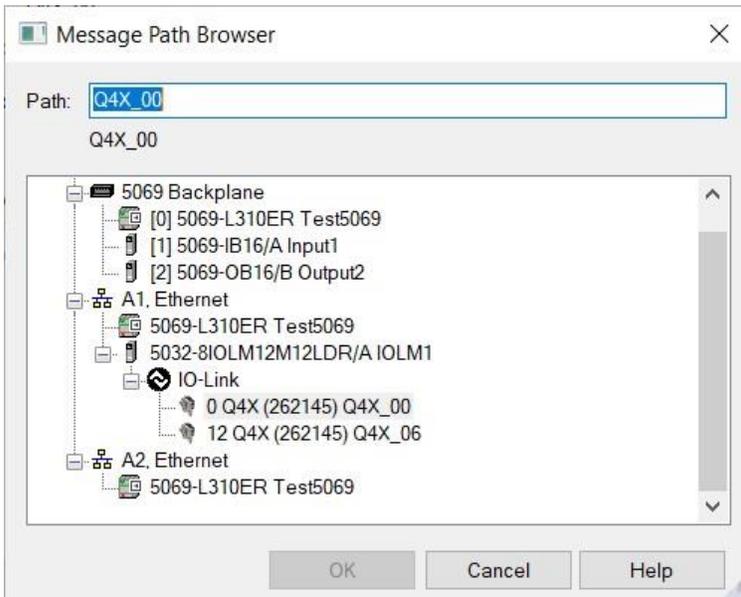
13. In the Message Configuration window, keep the Message Type “CIP Generic” and the Service Type “Custom”. Enter a Service Code of 33.

14. For the Source Element field, select “IOLM1.Message_Source_Data”.

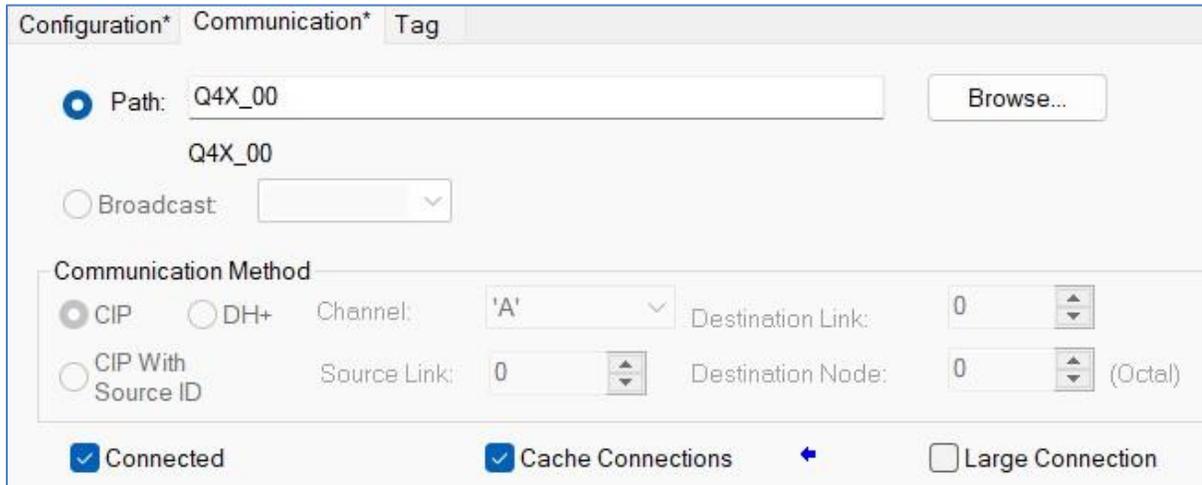


15. Now click on the Communication tab, then click the Browse button.

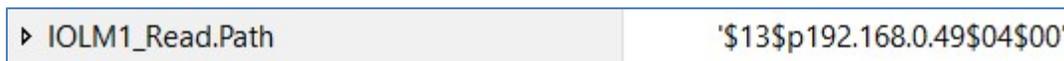
16. Select the IO-Link Master and then a device attached to the master. Then click OK again to close the Message Configuration window.



- The last step is to select the Connected option. A check mark will appear for both Connected and Cache Connections. These are necessary to ensure the write commands work correctly to the 5032. If these are not selected if a Write operation is done an error will occur.



- Press OK to finalize the changes.
- Now it is necessary to generate the ISDU IP Paths the PLC needs for the read and write commands. The 5032 requires this information to correctly navigate to the various ports on the 5032. When the communication path was selected in step 17 that was to link to one port of the IO-Link. Since accessing all the ports is important for most applications these paths need to be figured out.
- Got to the Controller Tags and look for the IOLM1_Read tags (or the tag you created for Step 6).
- Expand the tag and look for the Path tag. The path shown here is the path that the PLC needs to use to access the port and channel on the IO-Link Master. In this example a 5069 PLC was being used. The 192.168.0.49 is the ip address for the IO-Link Master. The \$00 at the end is the channel address (Port 0 Channel 0 in this case) to be accessed. Everything else is based on the PLC being used. It is best to copy this string.



- After copying the string go to the ISDU_Path tag created in Step 5.
- If more than 1 port is going to be used copy that string into the required sections of the array. The modify the array numbers that represent the port/channel that has an IO-Link Device plug into it. The example will show all 8 ports being used.
 - Port 0 Channel 0 uses \$00
 - Port 1 Channel 2 uses \$02
 - Port 2 Channel 4 uses \$04
 - Port 3 Channel 6 uses \$06
 - Port 4 Channel 8 uses \$08
 - Port 5 Channel 10 uses \$I (lower case L).
 - Port 6 Channel 12 uses \$p
 - Port 7 Channel 14 uses \$OE

24. The completed array should look like the image below. Remember only the ports that have IO-Link devices plugged into them need the path.

▲ Path1	{...}
▶ Path1[0]	'\$13\$p192.168.0.49\$04\$00'
▶ Path1[1]	'\$13\$p192.168.0.49\$04\$02'
▶ Path1[2]	'\$13\$p192.168.0.49\$04\$04'
▶ Path1[3]	'\$13\$p192.168.0.49\$04\$06'
▶ Path1[4]	'\$13\$p192.168.0.49\$04\$08'
▶ Path1[5]	'\$13\$p192.168.0.49\$04\$I'
▶ Path1[6]	'\$13\$p192.168.0.49\$04\$p'
▶ Path1[7]	'\$13\$p192.168.0.49\$04\$0E'

25. Setup of the IO-Link 5032 AOI is complete at this point. It will need a Banner IO-Link Device AOI to work properly, however.

4. Linking the Master AOI to Device AOIs

Select and install one or more Banner device AOIs. The device AOIs then need to be linked to this IO-Link Master AOI to provide control over IO-Link settings. See the relevant Banner IO-Link device AOI documents for more information.

Set "IOLM1.Port_Data.Port_Controlled" bits to 1 for every port on the IO-Link Master with a connected Banner device and its corresponding Device Parameter Data AOI. In the example below, a single Banner device is connected to port 1. The specific bit used here matches the port's label on the IO-Link Master. Some masters start with port 0, some start with port 1, depending on the manufacturer. Save the program to store these changes in the database.

◀ IOLM1.Port_Data	{...}
◀ IOLM1.Port_Data.Port_Controlled	2
IOLM1.Port_Data.Port_Controlled.0	0
IOLM1.Port_Data.Port_Controlled.1	1
IOLM1.Port_Data.Port_Controlled.2	0
IOLM1.Port_Data.Port_Controlled.3	0
IOLM1.Port_Data.Port_Controlled.4	0
IOLM1.Port_Data.Port_Controlled.5	0
IOLM1.Port_Data.Port_Controlled.6	0
IOLM1.Port_Data.Port_Controlled.7	0
IOLM1.Port_Data.Port_Controlled.8	0

5. Using the Paired IO-Link Master and Device Parameter Data AOIs

The goal is to make the Banner device's IO-Link Index and Subindex values appear in PLC tag arrays as if it were an EtherNet/IP-speaking device. Reading from and writing to the Banner IO-Link device becomes as easy as changing tag values in the PLC. All the complicated work of translating from EtherNet/IP to IO-Link is handled automatically, behind the scenes.

When the program is downloaded to the PLC and the PLC goes into run mode, the IO-Link Master AOI performs a global read for each connected Banner device AOI. The Banner device AOI then creates an archive copy of all writeable parameters for that device. This archive is used to determine whether one of the writeable data tags has been changed. If so, the AOI automatically triggers the process of acyclic writing, using correctly formatted CIP generic message commands.

There are three methods for acyclic reading of Banner device Index and Subindex values.

1. The initial global read, as requested by the IO-Link Master AOI after the PLC program is downloaded and run.
2. Manually via the "Command" variable found in every Device Parameter Data AOI. The "Command" register can be used to force one-time read or write actions, as described in Appendix A of any Banner Device Parameter AOI guide.

Acyclic writes to the IO-Link device are handled by simply changing the relevant tag values in the device's "Write_Data" tag array.

Appendix A **IODD Device Integration**

The Device Description Installation Tool is used to add an IODD file to Studio 5000. The Device Description Installation Tool is accessed under the Tools tab.

Select Register a device description (files) option. Press the Next button.

Browse for the IODD file. If the IODD file hasn't been downloaded yet go to Banner Engineering and download the file.

Complete the process by selecting the next button a few times.

IODD file is now accessible in Studio 5000.

Appendix B Error Handling & AOI Resets

Whenever an error related to the read or write Message Commands buried inside the AOI occurs, the “Num_Error_MSGS” variable will increment by 1.

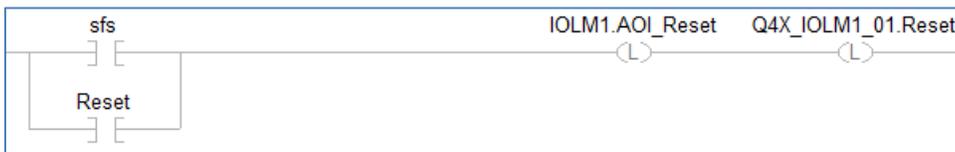
The specific error information will be stored in the “Error_Log” array. This array includes space for 10 errors. Each entry records whether the error occurred on a read (0) or write (1) attempt and which port on the IO-Link Master and Index on the IO-Link Device were involved. Once the error is logged, the AOI moves on to the next task. An example of an Error_Log entry is shown below, where an IO-Link Master AOI failed to write to Index 60 on the IO-Link device connected to port 1.

▲ IOLM1.Error_Log	{...}
▲ IOLM1.Error_Log[0]	{...}
▶ IOLM1.Error_Log[0].RW	1
▶ IOLM1.Error_Log[0].Port	1
▶ IOLM1.Error_Log[0].Index	60

Potential causes for errors include incorrect setup of the Device or Master AOI (wrong port number for device, wrong Port_Controlled array for master, or incorrect settings for the Master message commands), having the sensor physically connected to the incorrect port on the Master, or having no power to the IO-Link Master.

The “AOI_Reset” variable is used to restart the AOI from scratch. To initiate this reset, write a “1” to this register. The reset will occur, then turn the variable back to “0” automatically.

Best practices suggest adding a rung to your ladder logic program that resets all IO-Link Master and Device Parameter AOIs on the first scan. The example below shows one IO-Link Master, called IOLM4, and one connected Q5X having their respective AOIs being reset in this way.



Appendix C Halt AOI Operation

At times it may be desirable to halt the IO-Link Master AOI. This is especially true if you are using AOIs made by other manufacturers, particularly those made by the manufacturer of the IO-Link Master itself. The “Halt_Operation” variable can be used to stop the action of the Banner IO-Link Master AOI, allowing other AOIs to function correctly. When the other AOIs are done, the Banner IO-Link Master AOI can be reactivated.

▾ IOLM1	{...}		Banner_IOLM_v4
▸ IOLM1.Message_Source_Data	{...}	Decimal	SINT[190]
▸ IOLM1.Message_Destination_Data	{...}	Decimal	SINT[190]
▸ IOLM1.Error_Log	{...}		Banner_IOLM_EL_v2[10]
IOLM1.Error_Write_Retry	0	Decimal	BOOL
▸ IOLM1.Num_Error_MSGS	0	Decimal	DINT
IOLM1.IO_Link_Master_Busy	0	Decimal	BOOL
IOLM1.AOI_Reset	0	Decimal	BOOL
▸ IOLM1.Port_Data	{...}		Banner_IOL_Port_v4
IOLM1.Halt_Operation	0	Decimal	BOOL
IOLM1.AOI_Halted	0	Decimal	BOOL