



Turck IO-Link Master Add-On Instruction Guide, v2.1 3/26/2019

This document covers the installation and use of an 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 a Turck IO-Link Master. This AOI has four User Defined Tag data types.

This IO-Link Master AOI is meant to be used alongside one or more v2.1 Banner IO-Link Device Parameter Data AOIs.

This document was written using a Turck TBEN-L5-8IOL IO-Link Master. Other Turck IO-Link Masters will work as well.

Components

Banner_IOLM_T_v2.L5X
IOLService.L5X (an AOI from Turck)

NOTE:

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

It is intended to be linked to one or more v2.1 Banner Device Parameter AOIs to function.

UDT's Packaged with the AOI

Banner_IOLM_v2
Banner_IOLM_EL_v2
Banner_IOL_Port_v2
IOLRequest

Usage

Add and configure the relevant v2.1 Banner IO-Link Master AOI in your ladder logic program first; then add and configure v2.1 Banner IO-Link Device AOIs as desired, linking them to the Master AOI.

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.

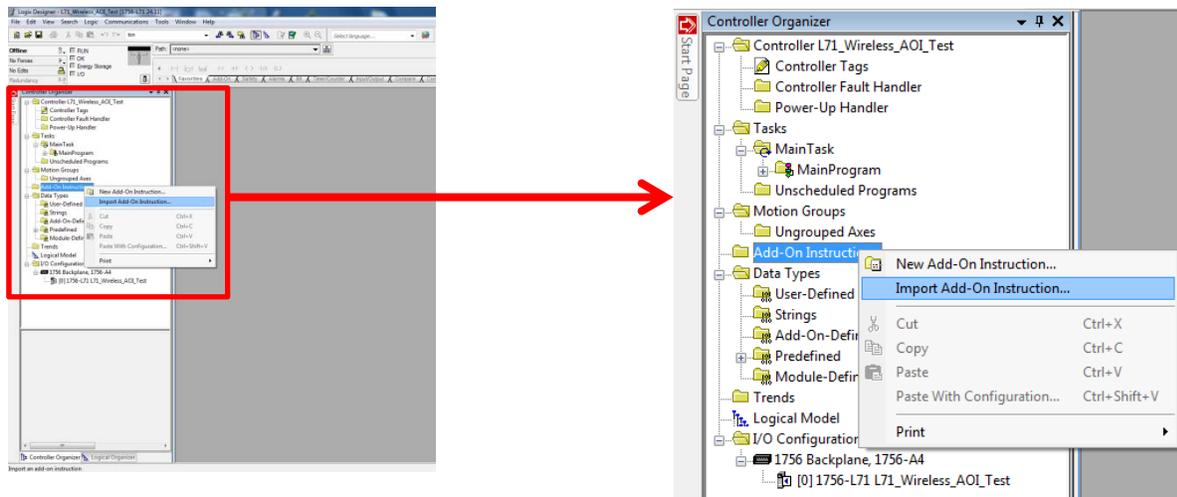
Contents

1. Installation Process	1
2. Configuring the Turck IO-Link Master	5
3. Configuring the IOLService AOI	6
4. Configuring the Banner IO-Link Master AOI	13
5. Linking the Master AOI to Device AOIs.....	20
6. Using the Paired IO-Link Master and Device Parameter Data AOIs.....	21
Appendix A Error Handling & AOI Resets.....	22
Appendix B Halt AOI Operation.....	23

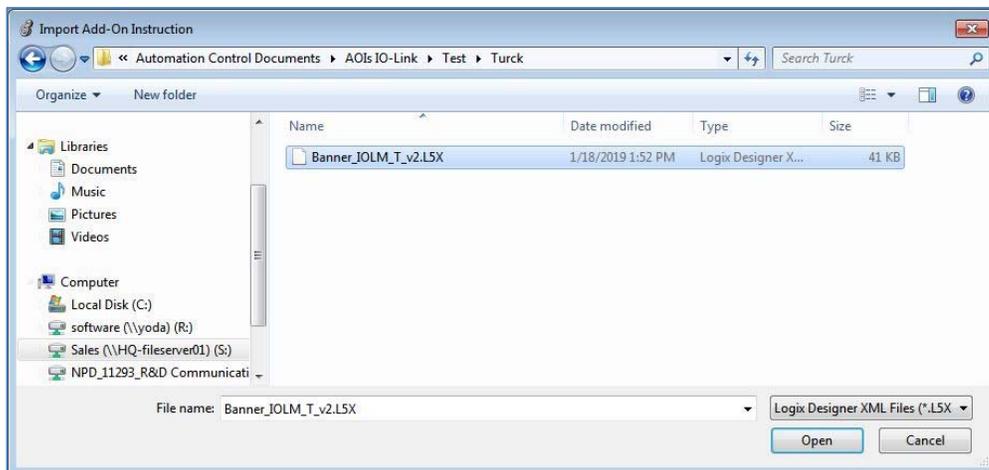
1. Installation Process

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

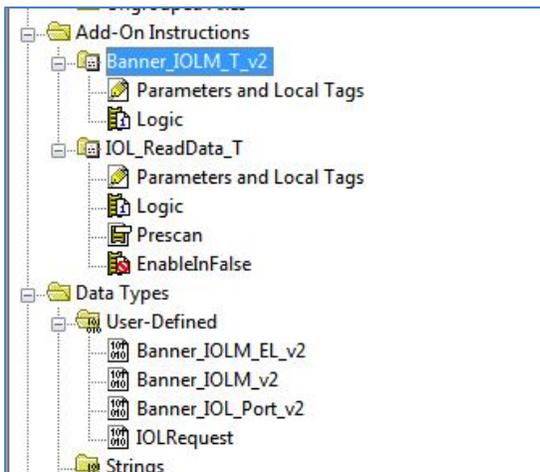
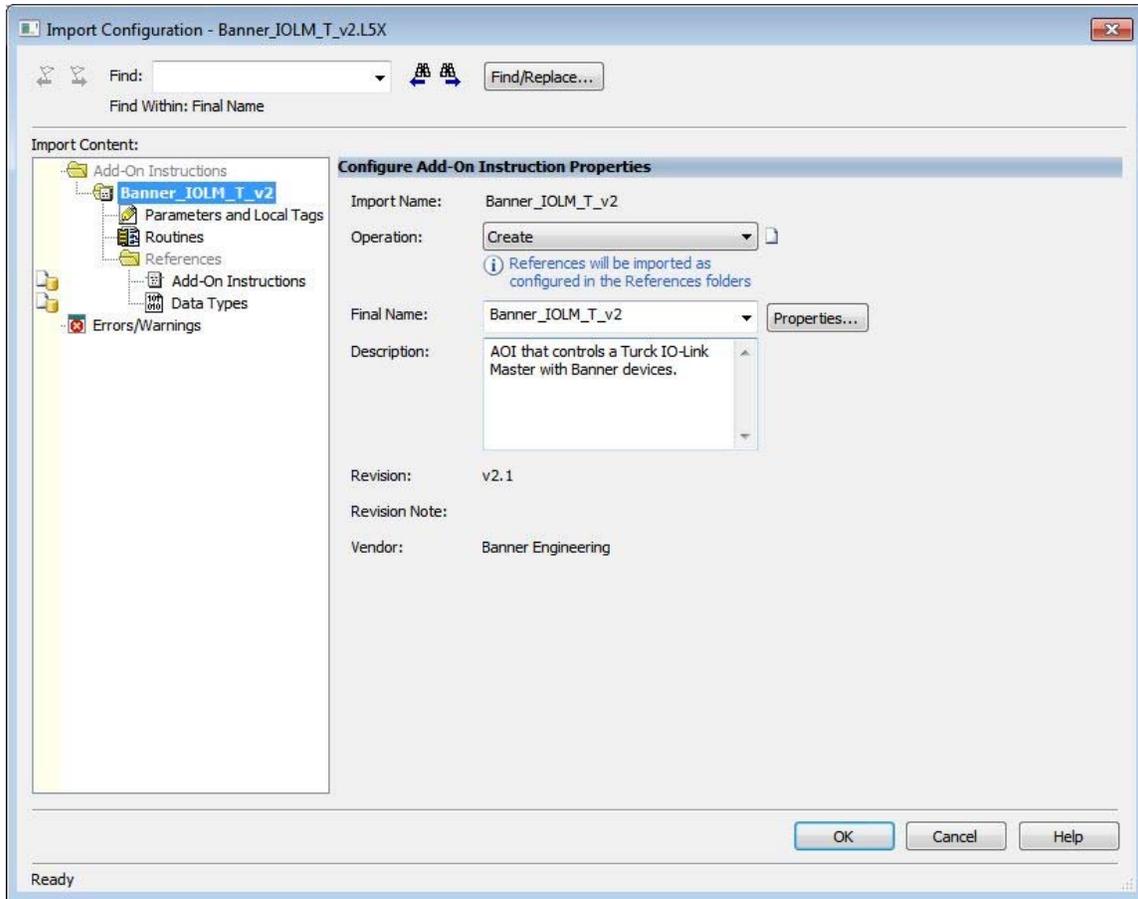
1. Open up a project.
2. In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.



3. Navigate to the correct file location and select the AOI to be installed. In this example the "Banner_IOLM_T_v2.L5X" file will be selected. Click the Open button.

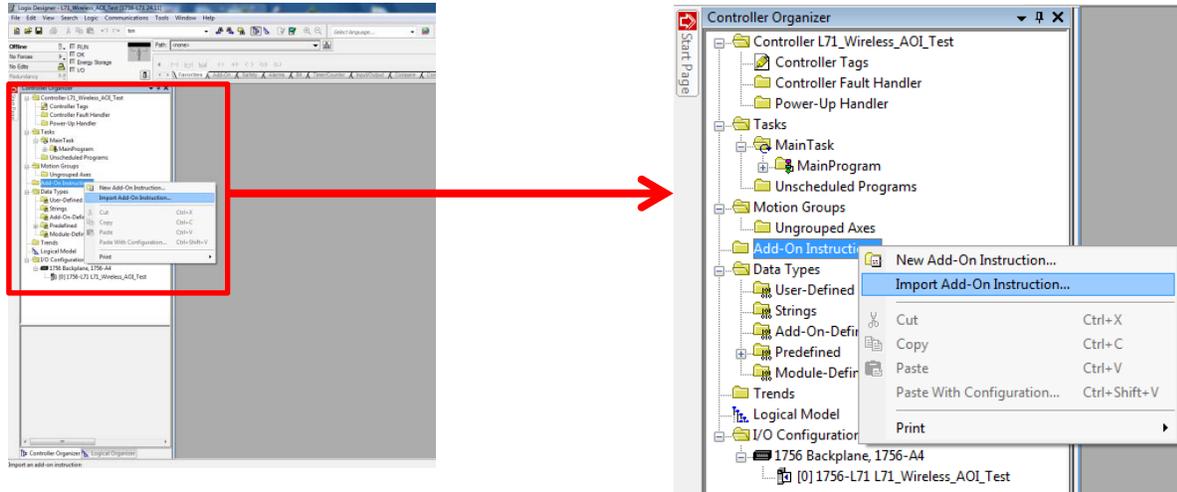


- 4. The Import Configuration window will pop up. The default selection will create all of the necessary items for the AOI. Click the OK button to complete the import process.

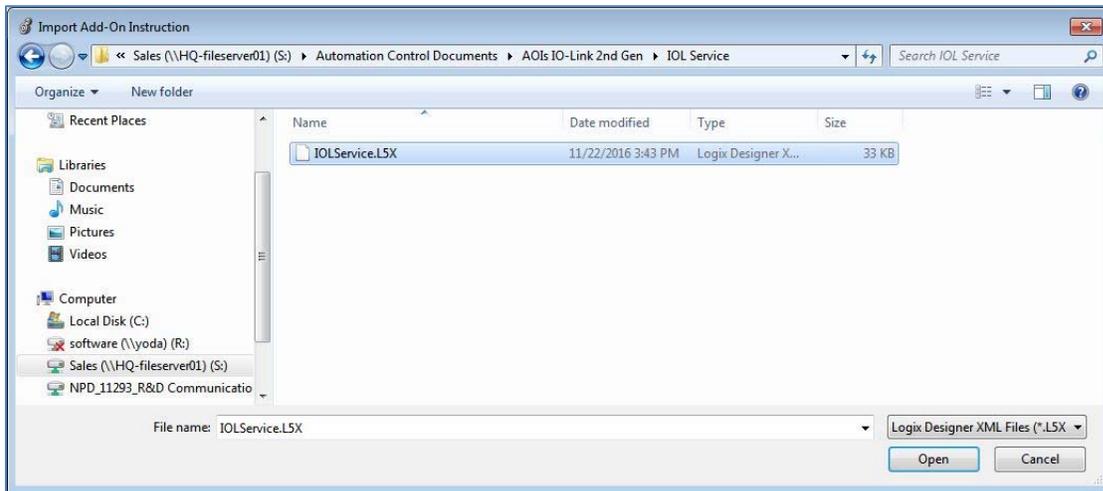


- 5. The AOI is added to the Controller Organizer window and should look similar to the picture at left.

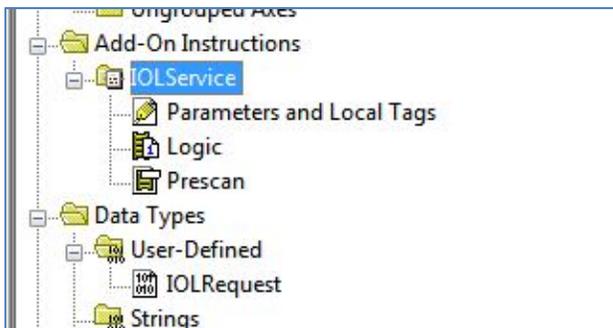
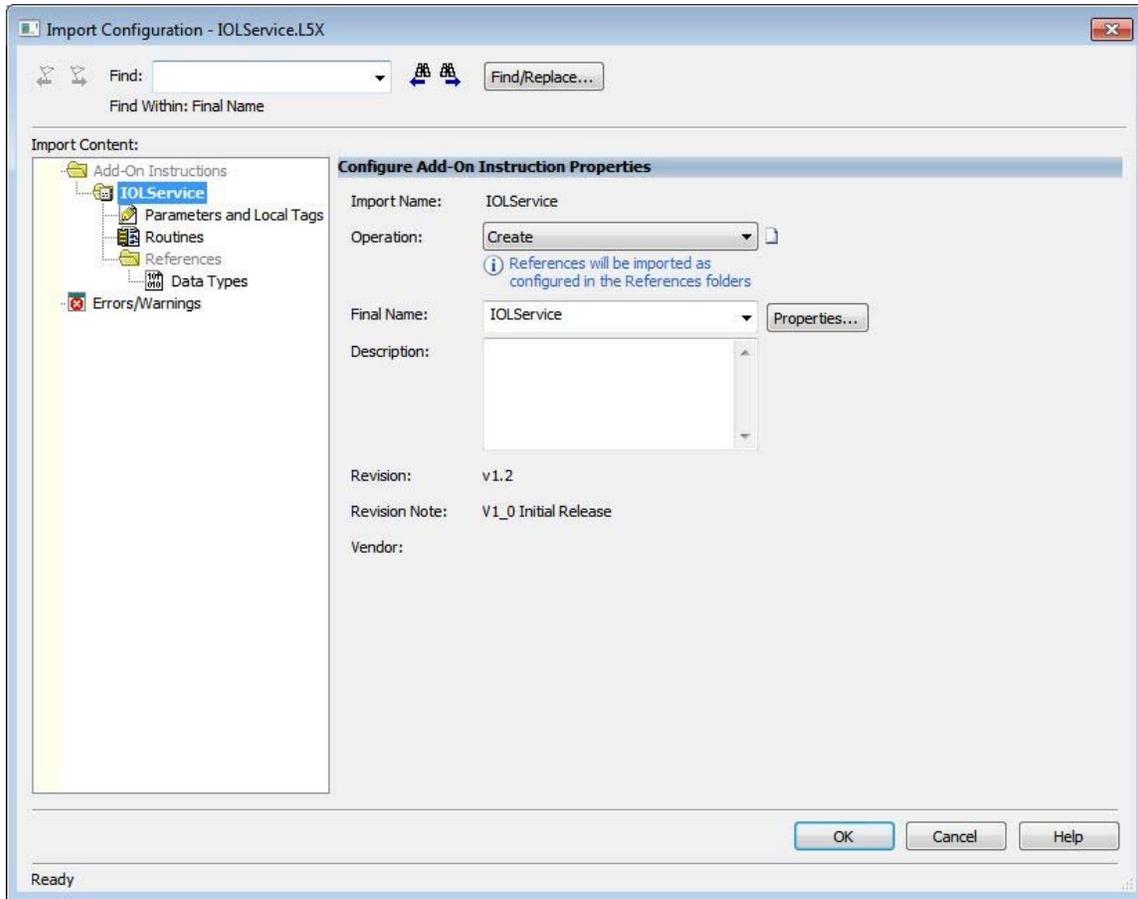
- In the Controller Organizer window, right-click on the Add-On Instruction folder. Select the Import Add-On Instruction option.



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



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



- 9. The AOI is added to the Controller Organizer window and should look similar to the picture at left.

- 10. AOI installation into the Logix Designer software complete.

2. Configuring the Turck IO-Link Master

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

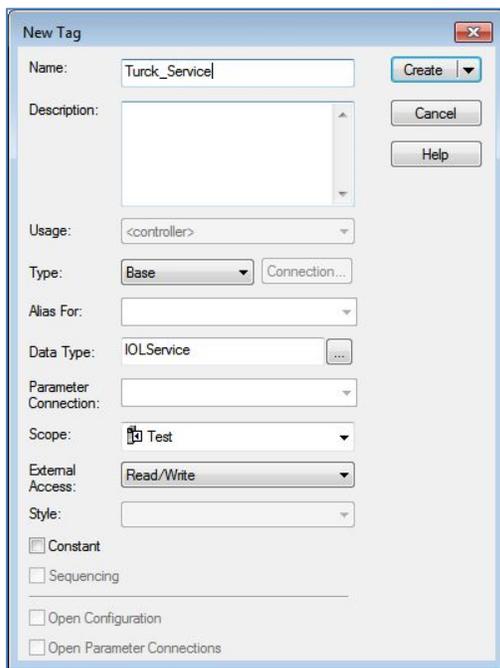
Create an Ethernet communications module for the Turck IO-Link Master device. In this example the EDS file was used, and the connection was named "IOLM4". The controller tags include Input (I) and Output (O) Assembly Instances. Each Assembly has a corresponding tag array. Creating this Class 1 EtherNet/IP implicit IO connection will provide the PLC access to the IO-Link sensor Process Data. Each port on the IO-Link Master is given a dedicated group of I and O registers. See the relevant Turck User's Guide for more information. This connection will also provide a communications pathway for the explicit messages used by the AOI to send IO-Link information to and from the Banner devices.

3. Configuring the IOLService AOI

1. Add the “IOLService” 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 array. The AOI includes a new type of User Defined Tag (UDT): a custom array of tags meant specifically for this AOI.



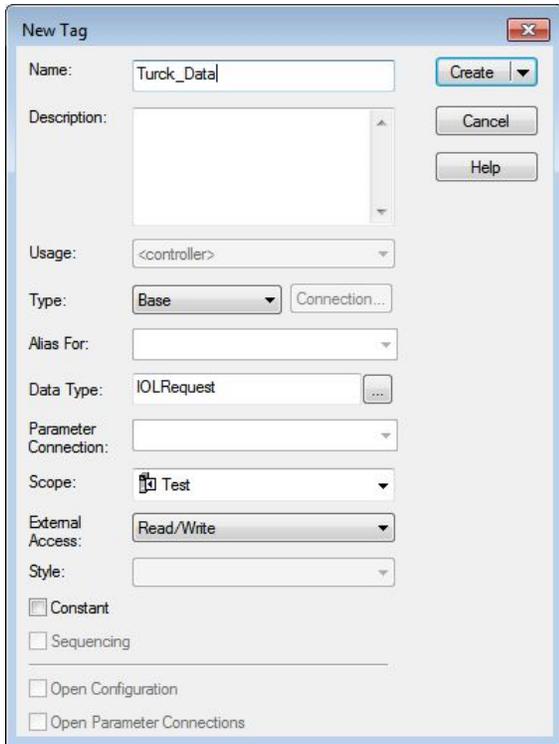
2. In the AOI, right-click on the question mark on the line labeled “IOLService”. Click New Tag. In this example, we’ll use the name “Turck_Service”. An instance of “IOLService” AOI is required for each Turck IO-Link Master.



The “EnableIn” and “EnableOut” variables are ladder logic rung status bits automatically added to all AOIs.

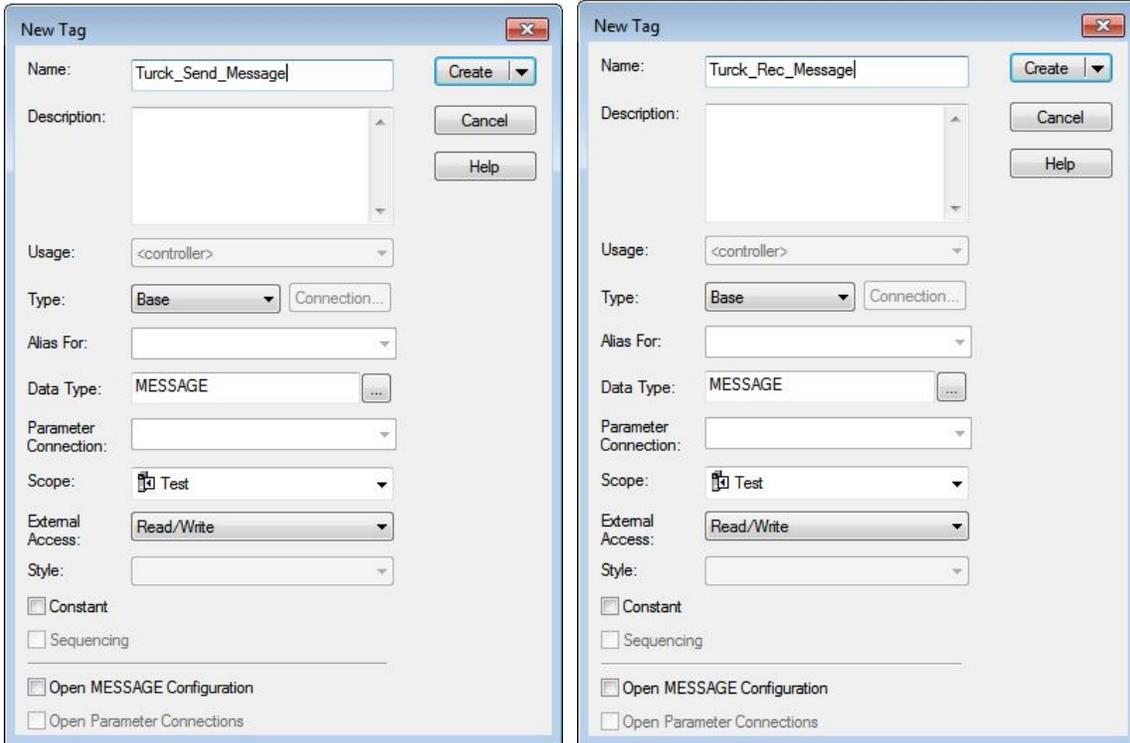
Turck_Service	{...}	{...}		IOLService
Turck_Service.EnableIn		1	Decimal	BOOL
Turck_Service.EnableOut		0	Decimal	BOOL

3. In the AOI, right-click on the question mark on the line labeled "IOLData". Click New Tag. In this example, we'll use the name "Turck_Data".



[-] Turck_Data		{ ... }	{ ... }		IOLRequest
[-] Turck_Data.Enable_Multiple_Nodes		0		Decimal	BOOL
[+] Turck_Data.ApplicationPath		''	{ ... }		STRING
[-] Turck_Data.Read		0		Decimal	BOOL
[-] Turck_Data.Write		0		Decimal	BOOL
[-] Turck_Data.Reset		0		Decimal	BOOL
[-] Turck_Data.Done		0		Decimal	BOOL
[-] Turck_Data.Error		0		Decimal	BOOL
[+] Turck_Data.Slot		0		Decimal	INT
[+] Turck_Data.Port		0		Decimal	INT
[+] Turck_Data.Index		0		Decimal	INT
[+] Turck_Data.SubIndex		0		Decimal	INT
[+] Turck_Data.Length		0		Decimal	INT
[+] Turck_Data.Handle		0		Decimal	INT
[+] Turck_Data.Status		0		Decimal	INT
[+] Turck_Data.NextHandle		0		Decimal	INT
[+] Turck_Data.ReceiveData		{ ... }	{ ... }	Decimal	SINT[145]
[+] Turck_Data.SendData		{ ... }	{ ... }	Decimal	SINT[145]
[+] Turck_Data.WriteData		0		Decimal	DINT
[+] Turck_Data.ActivePath		''	{ ... }		STRING

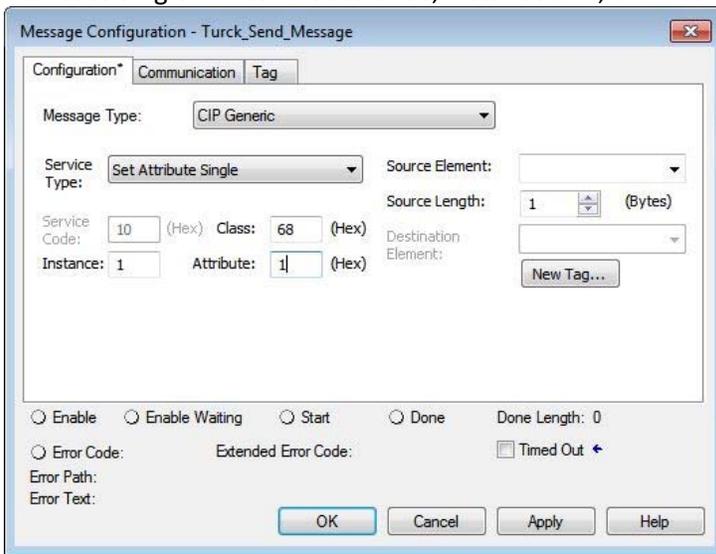
- Now we set up the two Messages used to read and write to devices connected to this IO-Link Master. Right click on the question mark for the "IOL_SendMSG" line in the AOI and choose New Tag. In this example we'll use the tag name "Turck_Send_Message". Click Create. Click on the line labeled "IOL_ReceiveMSG", choose New Tag, and name this one "Turck_Rec_Message". Click Create.



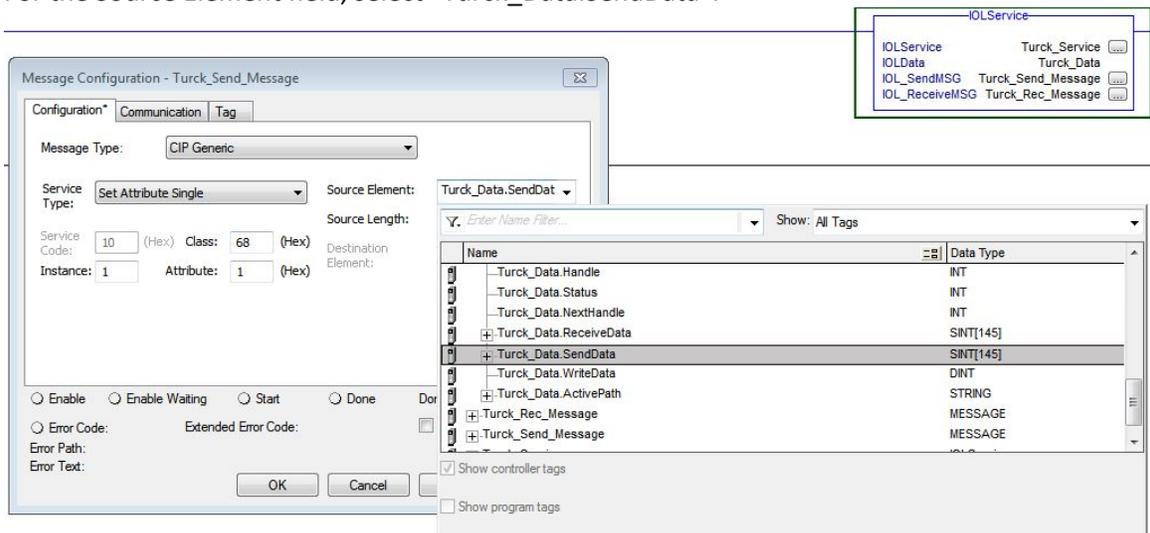
- Now configure “Turck_Send_Message”, setting up the Explicit Message that will handle communications between the PLC and the IO-Link Master. Click on the “...” button at the far right of the “IOL_SendMSG” line.



- In the Message Configuration window, keep the Message Type “CIP Generic” and the Service Type “Set Attribute Single”. Enter a Class of 68, Instance of 1, and Attribute of 1.



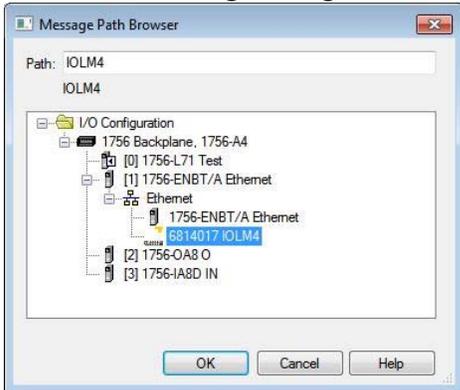
- For the Source Element field, select “Turck_Data.SendData”.



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



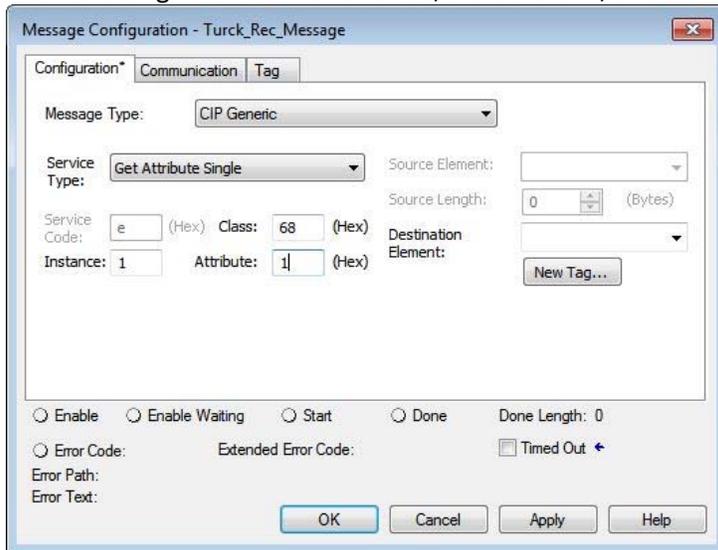
- Select the IO-Link Master, then click OK to close the Message Path Browser window, then click OK again to close the Message Configuration window.



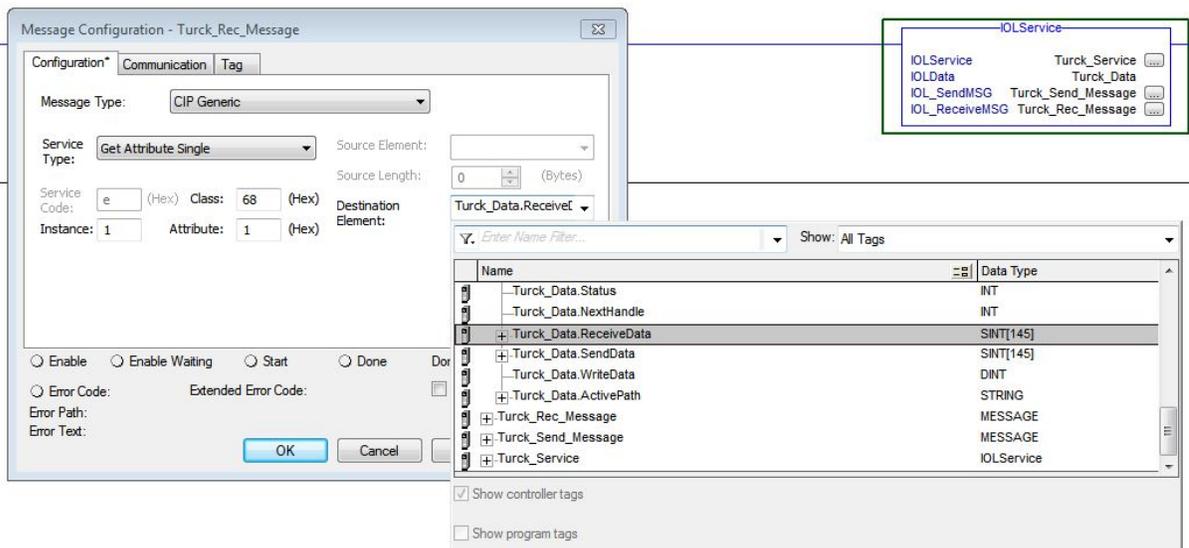
- Now configure “Turck_Rec_Message”, setting up the Explicit Message that will handle communications between the PLC and the IO-Link Master. Click on the “...” button at the far right of the “IOL_ReceiveMSG” line.



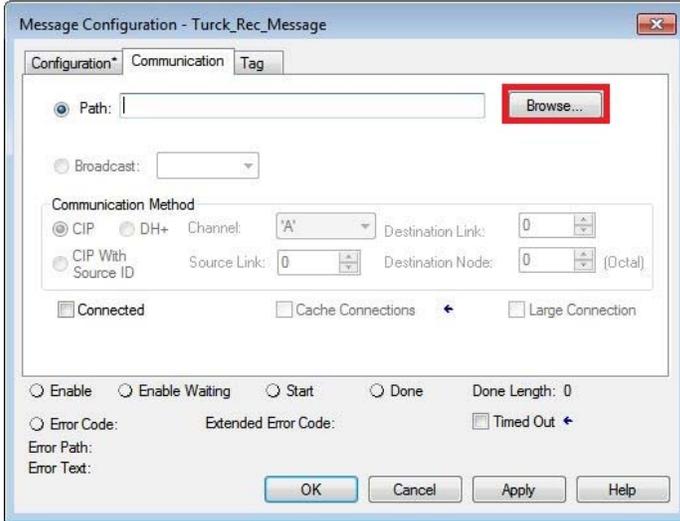
- In the Message Configuration window, keep the Message Type “CIP Generic” and the Service Type “Get Attribute Single”. Enter a Class of 68, Instance of 1, and Attribute of 1.



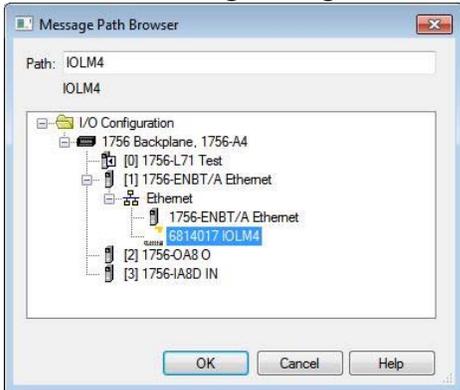
- For the Destination Element field, select “Turck_Data.ReceiveData”.



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



14. Select the IO-Link Master, then click OK to close the Message Path Browser window, then click OK again to close the Message Configuration window.

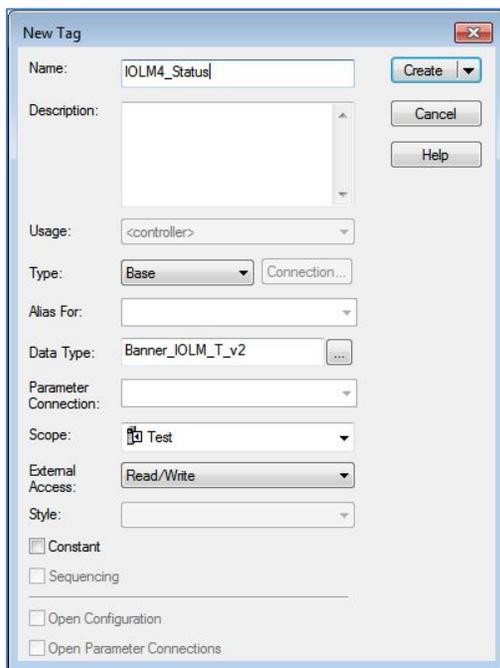


4. Configuring the Banner IO-Link Master AOI

1. Add the “Banner_IOLM_T_v2” 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 array. 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_T_v2”. Click New Tag. In this example, we’ll use the name “IOLM4_Status”. The example naming convention accounts for this being the #4 IO-Link Master in our program. More masters could be named IOLM1, IOLM3, etc.



The “EnableIn” and “EnableOut” variables are ladder logic rung status bits automatically added to all AOIs.

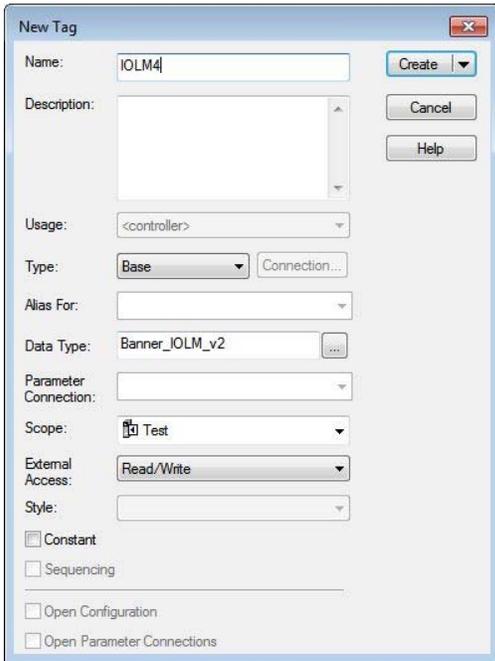
[-] IOLM4_Status		{...}	{...}		Banner_IOLM_T_v2
[-] IOLM4_Status.EnableIn		1		Decimal	BOOL
[-] IOLM4_Status.EnableOut		0		Decimal	BOOL
[+] IOLM4_Status.Slot		0		Decimal	INT

3. Now we set up the Message used to read and write to devices connected to this IO-Link Master. Right click on the question mark for the "Message" line in the AOI and choose New Tag. In this example we'll use the tag name "IOLM4_Message". Click Create.

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

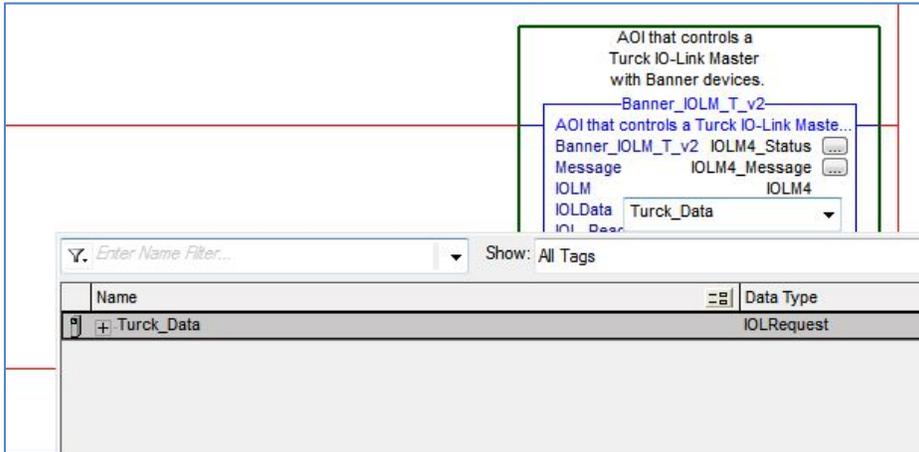
- Name: IOLM4_Message
- Description: (empty)
- Usage: <controller>
- Type: Base
- Alias For: (empty)
- Data Type: MESSAGE
- Parameter Connection: (empty)
- Scope: Test
- External Access: Read/Write
- Style: (empty)
- Constant:
- Sequencing:
- Open MESSAGE Configuration:
- Open Parameter Connections:

- Now create a new tag array for the "IOLM" line in the AOI. Here we used the name "IOLM4". 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.

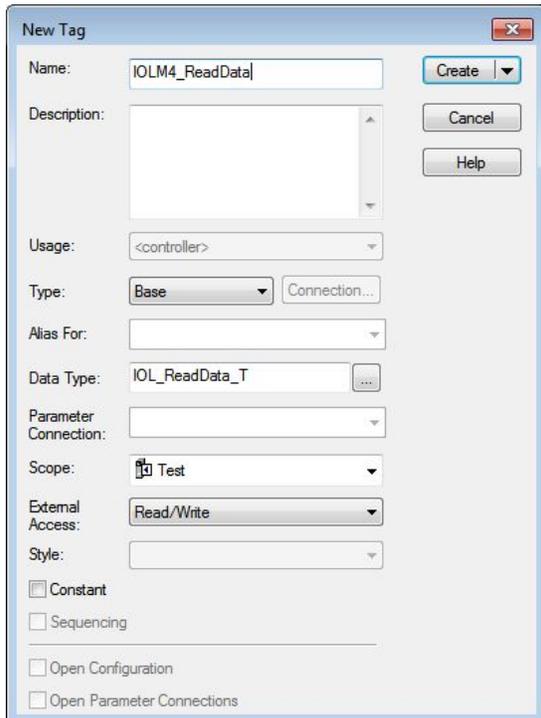


IOLM4	{ ... }	{ ... }		Banner_IOLM_v2
+ IOLM4.Message_Source_Data	{ ... }	{ ... }	Decimal	SINT[70]
+ IOLM4.Message_Destination_Data	{ ... }	{ ... }	Decimal	SINT[70]
+ IOLM4.Error_Log	{ ... }	{ ... }		Banner_IOLM_EL_v2[10]
- IOLM4.Error_Write_Retry		0	Decimal	BOOL
+ IOLM4.Num_Error_MSGS		0	Decimal	DINT
- IOLM4.IO_Link_Master_Busy		0	Decimal	BOOL
- IOLM4.AOI_Reset		0	Decimal	BOOL
+ IOLM4.Port_Data	{ ... }	{ ... }		Banner_IOLM_Port_v2
- IOLM4.Halt_Operation		0	Decimal	BOOL
- IOLM4.AOI_Halted		0	Decimal	BOOL

- 5. For the line labeled "IOLData" we do not create a new Tag Array. In this case we link the "Banner_IOLM_T_v2" AOI to a tag array in the "IOLService" AOI instance also tied to this Turck IO-Link Master. In our example we link to the IOLService AOI tag entitled "Turck_Data".



- 6. Now create a new tag array for the "IOL_ReadData" line in the AOI. Here we used the name "IOLM4_ReadData". Click Create.



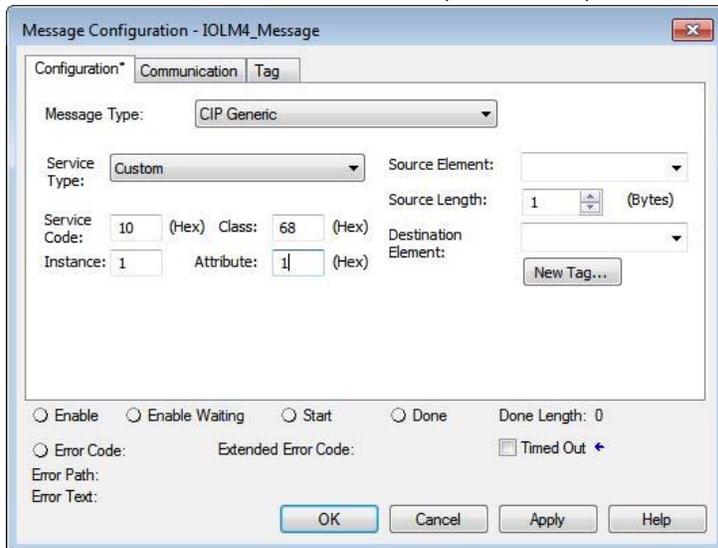
- The last line of the AOI, entitled "Slot", allows for the use of either I/O block (e.g. TBEN style) or distributed I/O style (e.g. BL67) IO-Link Masters. For this example, using a TBEN-L5-8IOL, the slot number is set to "1".



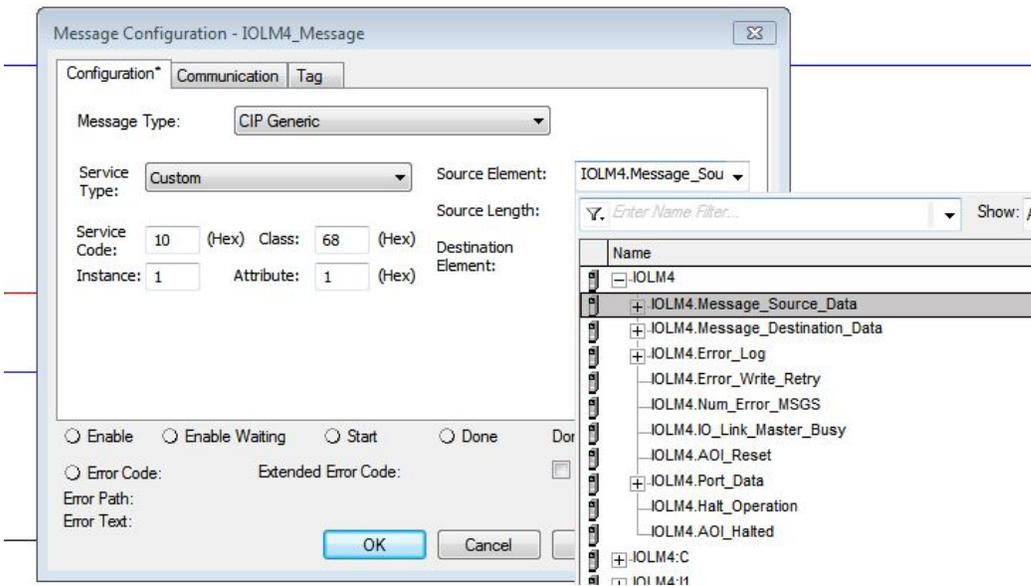
- Now configure "Message", setting up the Explicit Message that will handle communications between the PLC and the IO-Link Master. Click on the "..." button at the far right of the "Message" line.



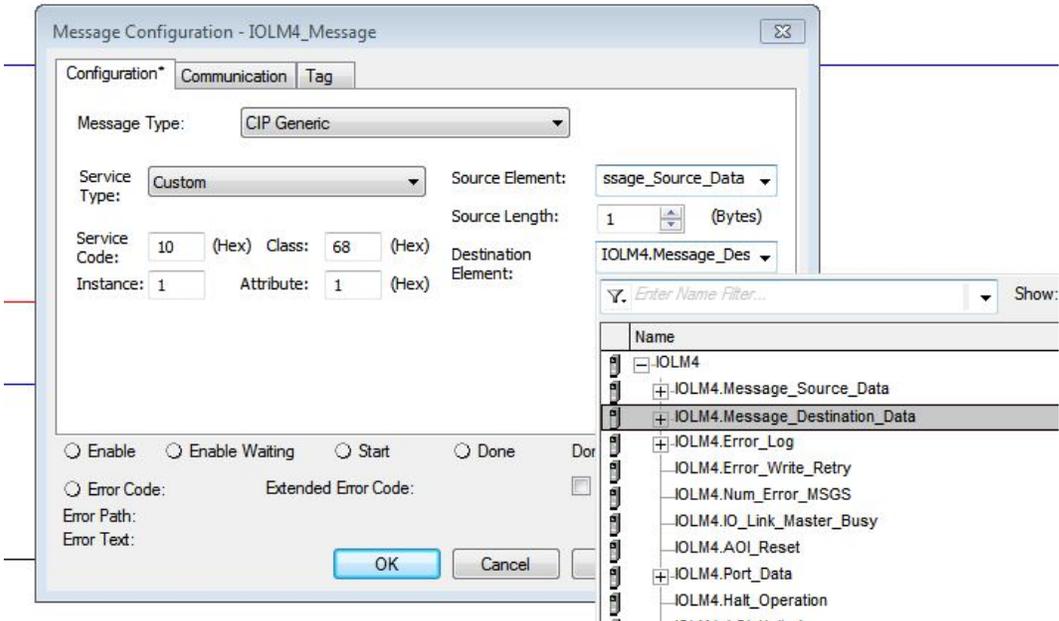
- In the Message Configuration window, keep the Message Type "CIP Generic" and the Service Type "Custom". Enter a Service Code of 10, Class of 68, Instance of 1, and Attribute of 1.



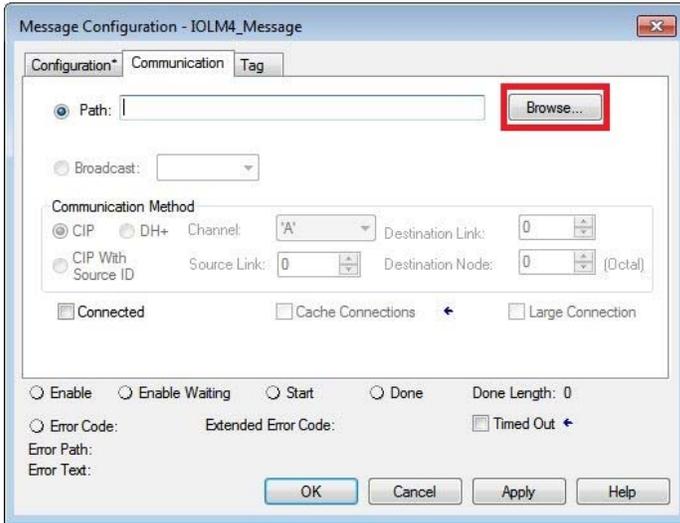
10. For the Source Element field, select "IOLM4.Message_Source_Data".



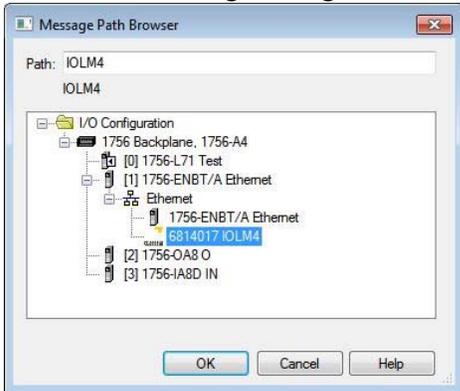
11. For Destination Element, select "IOLM4.Message_Destination_Data".



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



13. Select the IO-Link Master, then click OK to close the Message Path Browser window, then click OK again to close the Message Configuration window.



5. 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 “IOLM4.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 Q5X is connected to port 8. 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 manufacturer. Save the program to store these changes in the database.

IOLM4.Port_Data	{...}	{...}		Banner_IOL_Port_v2
IOLM4.Port_Data.Port_Controlled	256		Decimal	INT
IOLM4.Port_Data.Port_Controlled.0	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.1	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.2	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.3	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.4	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.5	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.6	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.7	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.8	1		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.9	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.10	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.11	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.12	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.13	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.14	0		Decimal	BOOL
IOLM4.Port_Data.Port_Controlled.15	0		Decimal	BOOL

6. 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 writable parameters for that device. This archive is used to determine whether one of the writable 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 toggling the "Port_Data.Device_Read" bits performs a one-time read of all values from a given Banner device AOI connected to a specific port on the IO-Link Master. To initiate this one-time read, toggle the bit (0-15) in the "Port_Data.Device_Read" variable corresponding to the port number in question. The AOI will read the device parameters from that port once, then turn the bit back to 0 automatically.
3. 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 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 6.

[-] IOLM3.Error_Log	{...}	{...}		Banner_IOLM_EL_v2[10]
[-] IOLM3.Error_Log[0]	{...}	{...}		Banner_IOLM_EL_v2
+ IOLM3.Error_Log[0].RW	1		Decimal	SINT
+ IOLM3.Error_Log[0].Port	6		Decimal	SINT
+ IOLM3.Error_Log[0].Index	60		Decimal	SINT

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 B 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.

- IOLM3	{...}	{...}		Banner_IOLM_v2	U
+ IOLM3.Message_Source_Data	{...}	{...}	Decimal	SINT[70]	U
+ IOLM3.Message_Destination_Data	{...}	{...}	Decimal	SINT[70]	U
+ IOLM3.Error_Log	{...}	{...}		Banner_IOLM_EL_v2[10]	U
- IOLM3.Error_Write_Retry	0		Decimal	BOOL	U
+ IOLM3.Num_Error_MSGS	65		Decimal	DINT	U
- IOLM3.IO_Link_Master_Busy	0		Decimal	BOOL	U
- IOLM3.AOI_Reset	0		Decimal	BOOL	U
+ IOLM3.Port_Data	{...}	{...}		Banner_IOL_Port_v2	U
- IOLM3.Halt_Operation	0		Decimal	BOOL	U
- IOLM3.AOI_Halted	0		Decimal	BOOL	U