Integration of In-Sight with AB PLCs running RSLogix

Author: Samantha Frost

Published: August 11, 2017
Revision: 1.0

Cognex

Contents

Communicate with a Rockwell ControlLogix PLC

EtherNet/IP Communications

This section describes how to transfer data between an In-Sight vision system and a ControlLogix or CompactLogix PLC on an EtherNet/IP network, using Rockwell RSLogix 5000 software.

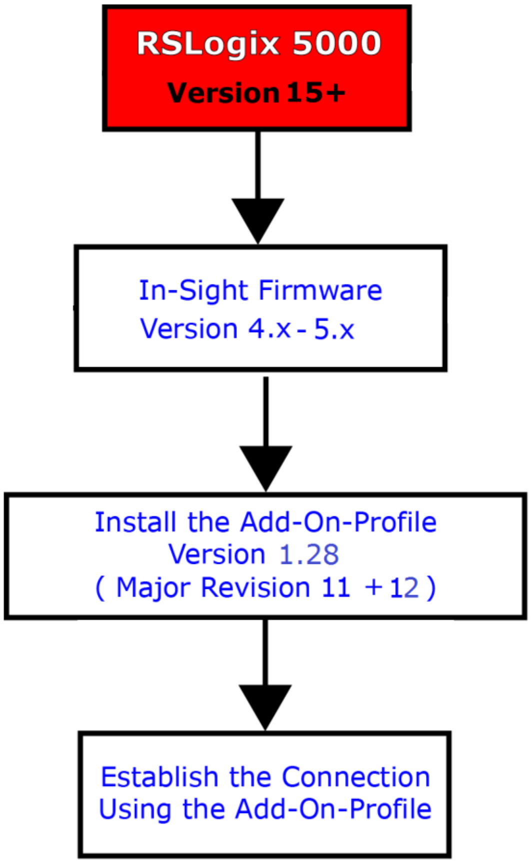
- Integration with RSLogix 5000
- EDS Generated Profile or Add-on Profile

Integration with RSLogix 5000

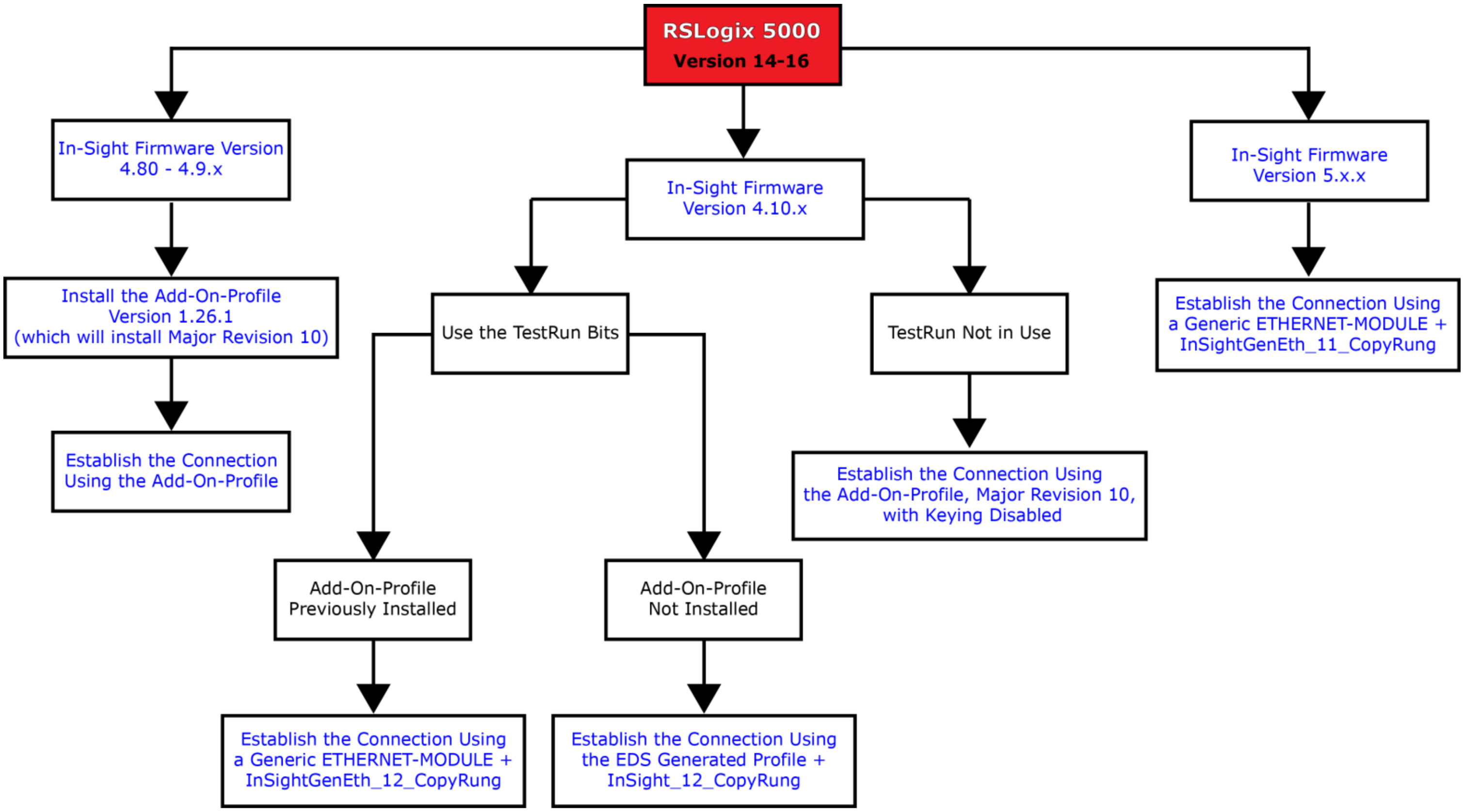
The steps to configure an implicit connection and transfer of data depend on the In-Sight firmware version installed to the vision system, and the version of RSLogix 5000 software.

Please note: It is mandatory that you have RSLogix 5000 version 15.00 or later in order to use AOP version 1.28 of the profile. This version of the profile is not compatible with earlier versions of RSLogix 5000. This version provides AOP support to the next generation models: IS8000 and IS7000 Gen 2; as well as the legacy In-Sight models.

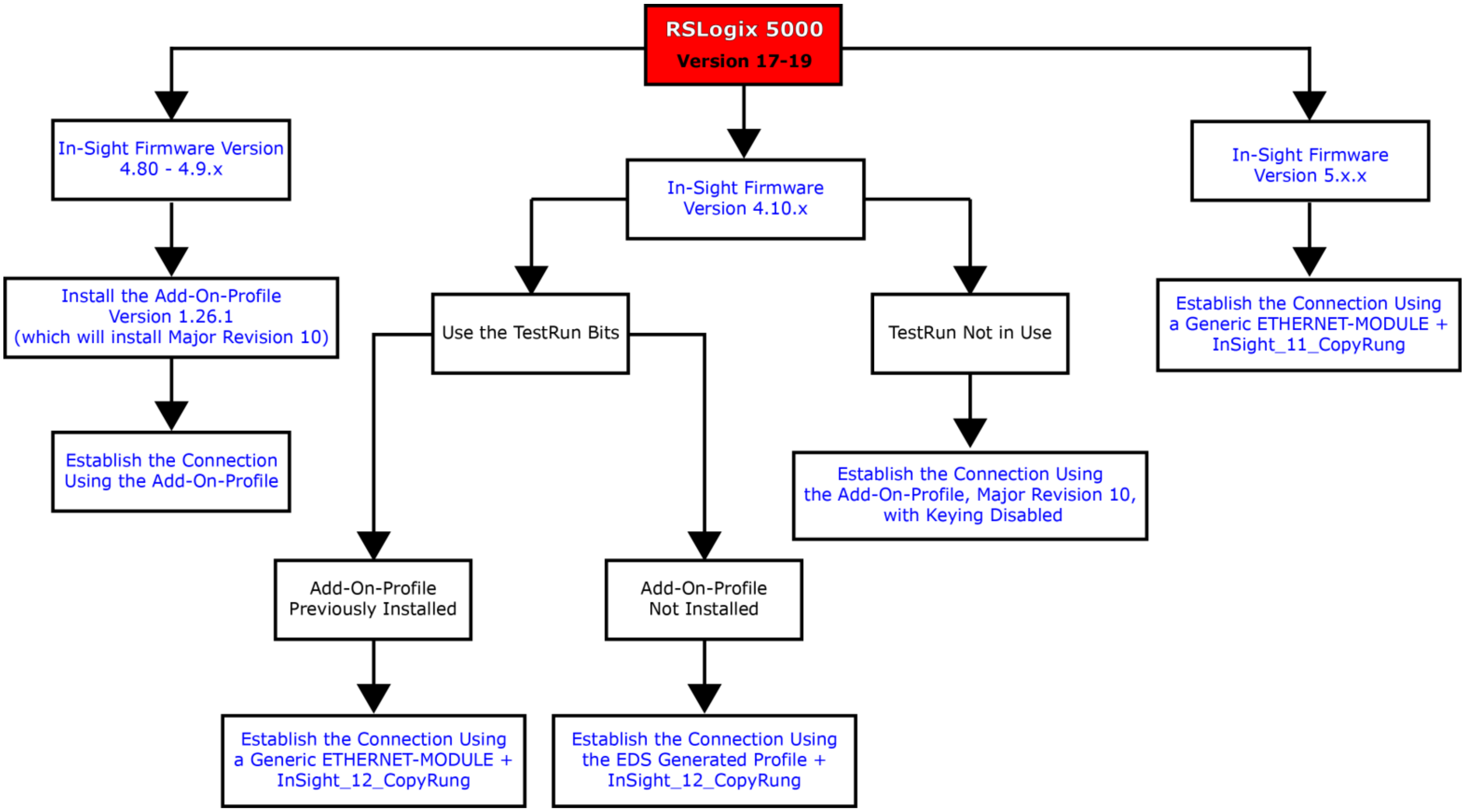
RSLogix 5000, Version 15+, using AOP version 1.28



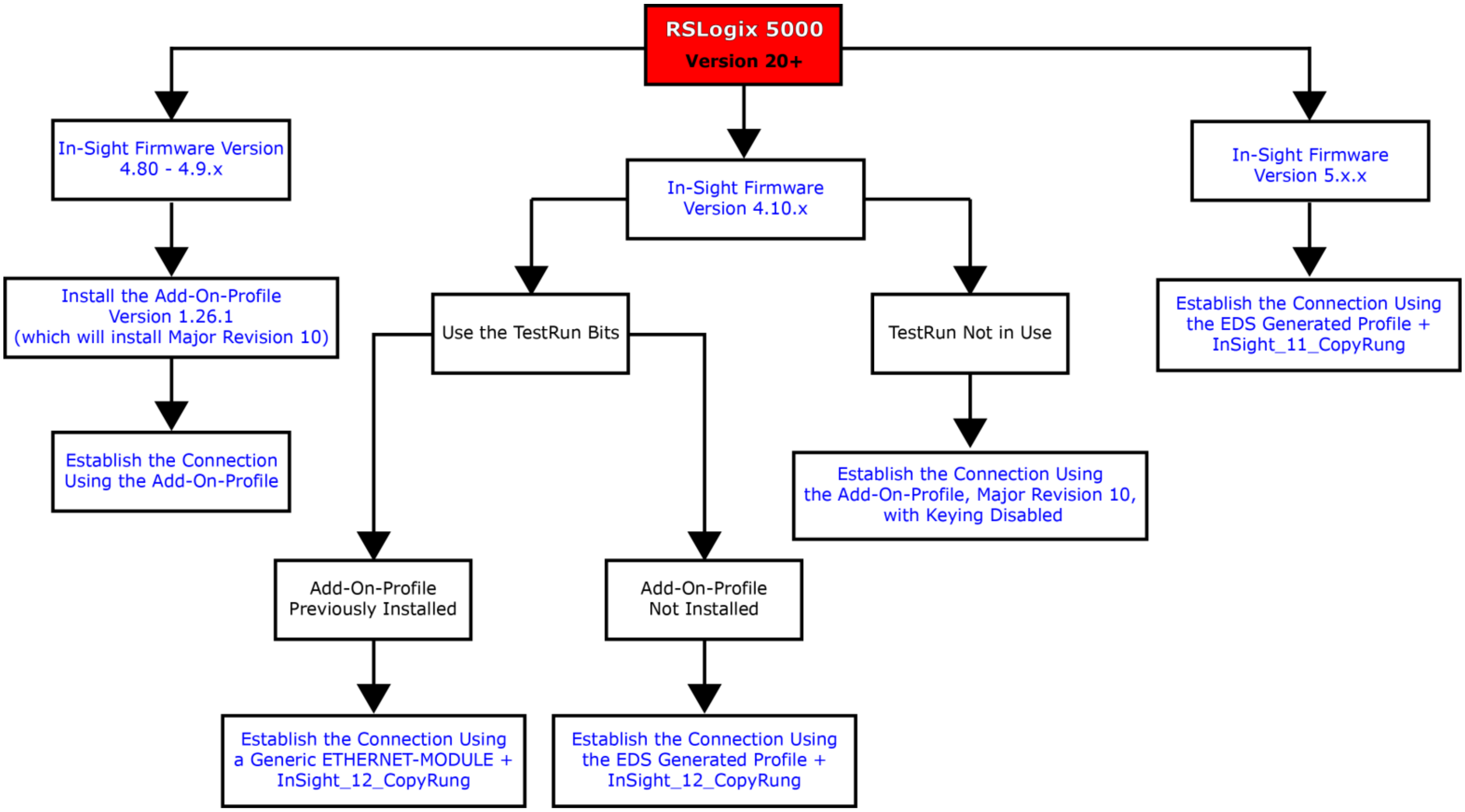
RSLogix 5000, Version 14 - 16, using Generic or EDS Generated profile:


RSLogix 5000
Version 14-16
In-Sight Firmware Version 4.80 - 4.9.x
Install the Add-On-Profile Version 1.26.1 (which will install Major Revision 10)
Establish the Connection Using the Add-On-Profile
In-Sight Firmware Version 4.10.x
Use the TestRun Bits
Add-On-Profile Previously Installed
Establish the Connection Using a Generic ETHERNET-MODULE + InSightGenEth_12_CopyRung
Add-On-Profile Not Installed
Establish the Connection Using the EDS Generated Profile + InSight_12_CopyRung
TestRun Not in Use
Establish the Connection Using the Add-On-Profile, Major Revision 10, with Keying Disabled
In-Sight Firmware Version 5.x.x
Establish the Connection Using a Generic ETHERNET-MODULE + InSightGenEth_11_CopyRung

RSLogix 5000, Version 17-19, using Generic or EDS Generated profile:


RSLogix 5000
Version 17-19
In-Sight Firmware Version 4.80 - 4.9.x
Install the Add-On-Profile Version 1.26.1 (which will install Major Revision 10)
Establish the Connection Using the Add-On-Profile
In-Sight Firmware Version 4.10.x
Use the TestRun Bits
Add-On-Profile Previously Installed
Establish the Connection Using a Generic ETHERNET-MODULE + InSight_12_CopyRung
Add-On-Profile Not Installed
Establish the Connection Using the EDS Generated Profile + InSight_12_CopyRung
TestRun Not in Use
Establish the Connection Using the Add-On-Profile, Major Revision 10, with Keying Disabled
In-Sight Firmware Version 5.x.x
Establish the Connection Using a Generic ETHERNET-MODULE + InSight_11_CopyRung

RSLogix 5000, Version 20+, using Generic or EDS Generated profile:

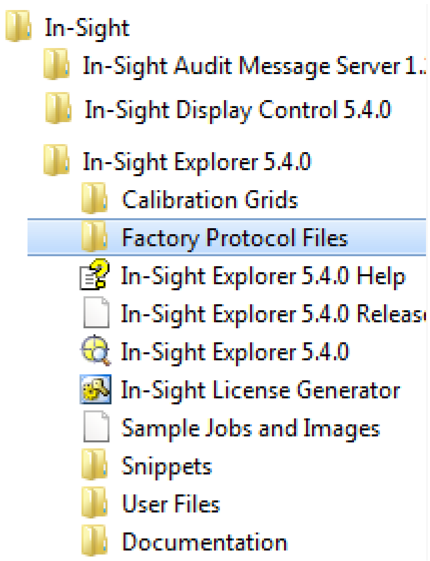

RSLogix 5000
Version 20+
In-Sight Firmware Version 4.80 - 4.9.x
Install the Add-On-Profile Version 1.26.1 (which will install Major Revision 10)
Establish the Connection Using the Add-On-Profile
In-Sight Firmware Version 4.10.x
Use the TestRun Bits
Add-On-Profile Previously Installed
Establish the Connection Using a Generic ETHERNET-MODULE + InSight_12_CopyRung
Add-On-Profile Not Installed
Establish the Connection Using the EDS Generated Profile + InSight_12_CopyRung
TestRun Not in Use
Establish the Connection Using the Add-On-Profile, Major Revision 10, with Keying Disabled
In-Sight Firmware Version 5.x.x
Establish the Connection Using the EDS Generated Profile + InSight_11_CopyRung

EDS Generated Profile or Add-on Profile

EDS Generated Profile

Electronic Data Sheets (EDS) are text files used by various network configuration tools to identify In-Sight vision systems on a network. EDS files can be used by any EtherNet/IP device and can be ported across many platforms and vendors.

In-Sight EDS files are installed with In-Sight Explorer software in the Factory Protocol Files folder:


In-Sight
In-Sight Audit Message Server 1.
In-Sight Display Control 5.4.0
In-Sight Explorer 5.4.0
Calibration Grids
Factory Protocol Files
In-Sight Explorer 5.4.0 Help
In-Sight Explorer 5.4.0 Releas
In-Sight Explorer 5.4.0
In-Sight License Generator
Sample Jobs and Images
Snippets
User Files
Documentation

These EDS files are then imported into the appropriate software. For installation information using the EDS or Generic Module, please refer to the *ESTABLISH THE CONNECTION USING THE EDS GENERATED PROFILE* or the *ESTABLISH THE CONNECTION USING A GENERIC ETHERNET-MODULE* section of this document.

Add-On Profile

The Add-On Profile is functionality added to RSLogix 5000 software and is used for communication with Rockwell PLCs. The user interface for Add-On Profiles provides enumerated access to the configuration and status data with full range checking. Data entered is validated to assure that selections made are correct and consistent with the desired setup. The Add-On Profile understands the structure of I/O and configuration data and creates tags using names assigned to the device in RSLogix 5000. There is complete enumeration of the tag to visibly differentiate the data and all data is available using the tag name.

The In-Sight Add-On Profile installer is available from the In-Sight support site. Current version is 1.28 and includes all In-Sight vision systems including the new next generation models (IS8000, IS7500-IS7900 series).

Notes:

- The Add-on Profile installs Major Revision 11 for In-Sight 2000, 57xx, 75xx-79xx, 8xxx, and 9xxx Series Systems, and Major Revision 12 for all legacy models (In-Sight 5xxx,70xx-74xx,10xx Series Systems).
- Major revision 12 adds TestRun bits to the legacy models.
- If the Add-On Profile is installed, the EDS generated profile is hidden for In-Sight vision systems with 4.x.x firmware installed.
- While the latest In-Sight Add-On Profile contains all the contents from previous revisions and will not break compatibility with existing applications, whether or not the connection will be established is tied to the Major Revision field in the Add-On Profile.

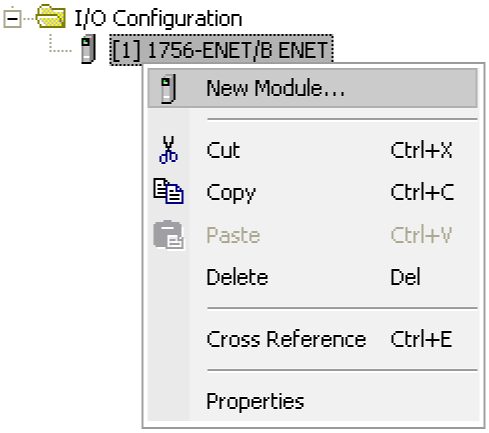
Establish the Connection Using the Add-On Profile

To establish an implicit messaging connection with a single ControlLogix PLC using the Add-On Profile:

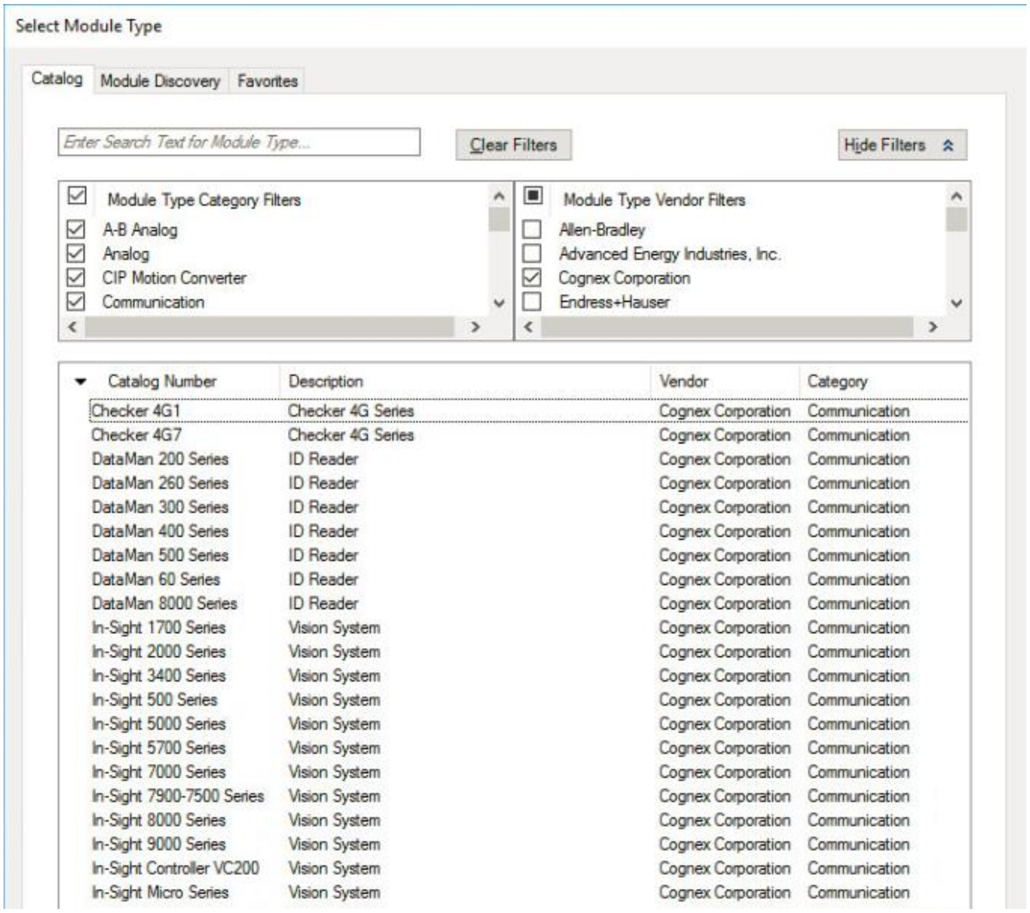
1. Open RSLogix 5000 and load the PLC's project.

   > **Note**: The PLC must be Offline to add connections in RSLogix 5000.

2. Under the I/O Configuration node, select the *Ethernet Node* under the Ethernet Module, right-click on the icon and select New Module from the menu:



3. When the following dialog appears, select your model of In-Sight Vision System from the list. This option will appear once the Add-On Profile is installed.

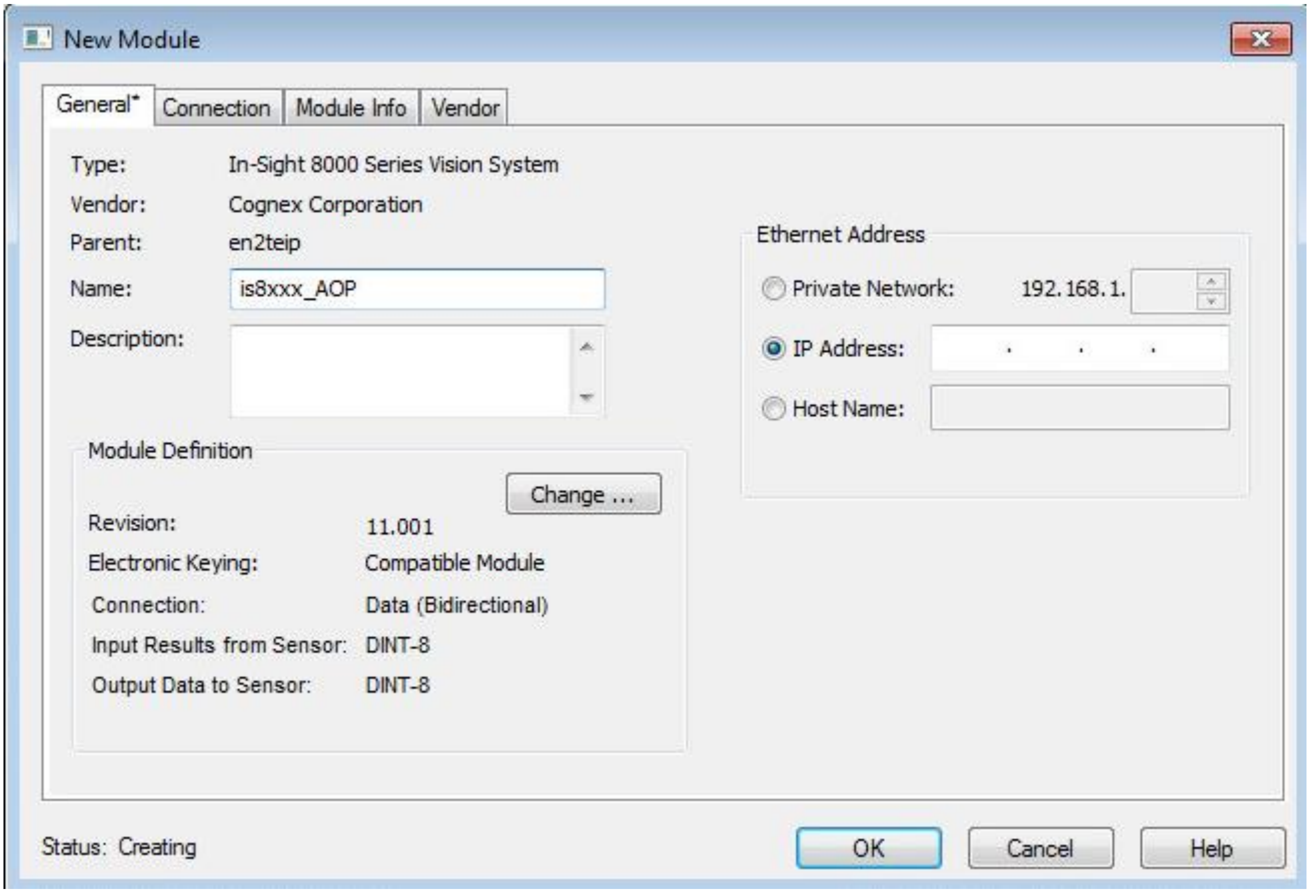


☑ Module Type Category Filters
☑ A-B Analog
☑ Analog
☑ CIP Motion Converter
☑ Communication

■ Module Type Vendor Filters
☐ Allen-Bradley
☐ Advanced Energy Industries, Inc.
☑ Cognex Corporation
☐ Endress+Hauser

Catalog Number	Description	Vendor	Category
Checker 4G1	Checker 4G Series	Cognex Corporation	Communication
Checker 4G7	Checker 4G Series	Cognex Corporation	Communication
DataMan 200 Series	ID Reader	Cognex Corporation	Communication
DataMan 260 Series	ID Reader	Cognex Corporation	Communication
DataMan 300 Series	ID Reader	Cognex Corporation	Communication
DataMan 400 Series	ID Reader	Cognex Corporation	Communication
DataMan 500 Series	ID Reader	Cognex Corporation	Communication
DataMan 60 Series	ID Reader	Cognex Corporation	Communication
DataMan 8000 Series	ID Reader	Cognex Corporation	Communication
In-Sight 1700 Series	Vision System	Cognex Corporation	Communication
In-Sight 2000 Series	Vision System	Cognex Corporation	Communication
In-Sight 3400 Series	Vision System	Cognex Corporation	Communication
In-Sight 500 Series	Vision System	Cognex Corporation	Communication
In-Sight 5000 Series	Vision System	Cognex Corporation	Communication
In-Sight 5700 Series	Vision System	Cognex Corporation	Communication
In-Sight 7000 Series	Vision System	Cognex Corporation	Communication
In-Sight 7900-7500 Series	Vision System	Cognex Corporation	Communication
In-Sight 8000 Series	Vision System	Cognex Corporation	Communication
In-Sight 9000 Series	Vision System	Cognex Corporation	Communication
In-Sight Controller VC200	Vision System	Cognex Corporation	Communication
In-Sight Micro Series	Vision System	Cognex Corporation	Communication

4. After the selection is made, the configuration dialog for the In-Sight Vision system will be displayed:



Type: In-Sight 8000 Series Vision System
Vendor: Cognex Corporation
Parent: en2teip
Name: is8xxx_AOP
Description:

Module Definition
Revision: 11.001
Electronic Keying: Compatible Module
Connection: Data (Bidirectional)
Input Results from Sensor: DINT-8
Output Data to Sensor: DINT-8

Ethernet Address
○ Private Network: 192.168.1.
● IP Address: . . .
○ Host Name:

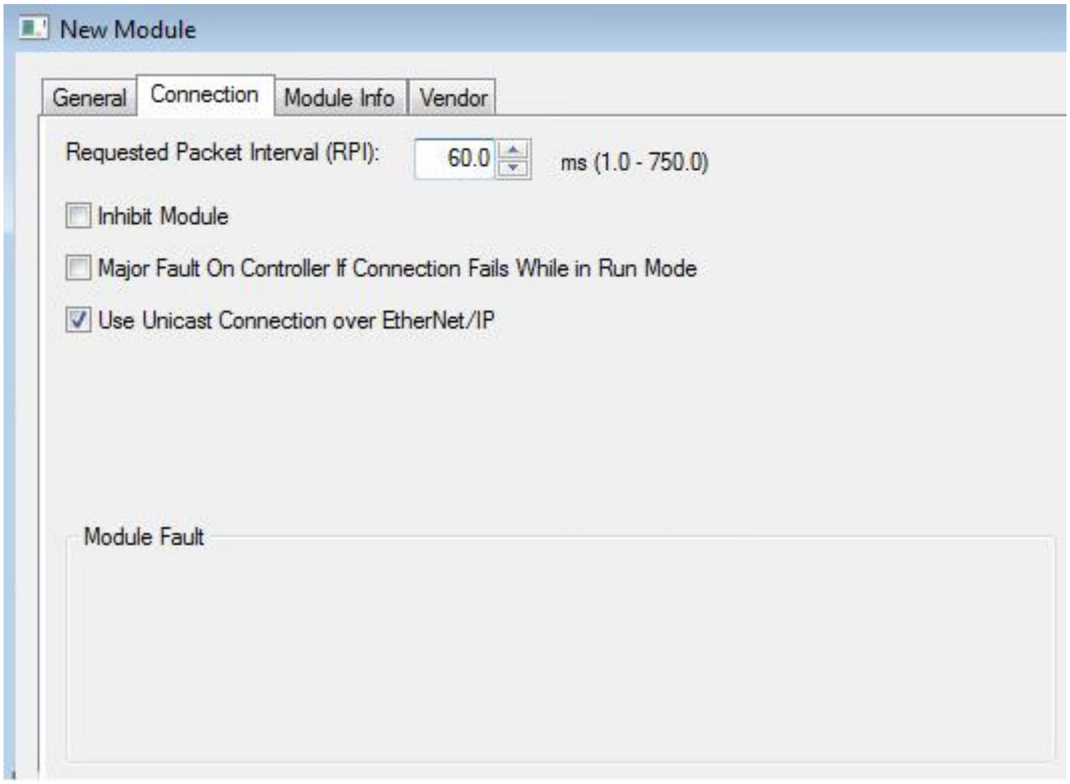
Status: Creating

> **Note**: The Time Sync tab is displayed only when connected to an In-Sight 5000, In-Sight 7500-7900 or In-Sight 9000 series vision systems.

The following fields need to be configured:

- **Name**: This is the name given to the In-Sight vision system; the tags created in RSLogix 5000 will be based on this name. It is recommended that the In-Sight vision system's name be used, to maintain consistency.
- **IP Address**: The IP address of the In-Sight vision system.
- **Host Name**: This setting is optional; only use this setting if there is a Domain Name Server (DNS) on the network.

5. After accepting the general communication parameters, the next step in the configuration process is displayed in the figure below:



Requested Packet Interval (RPI): 60.0 ms (1.0 - 750.0)

☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

☑ Use Unicast Connection over EtherNet/IP

The following fields can be configured:

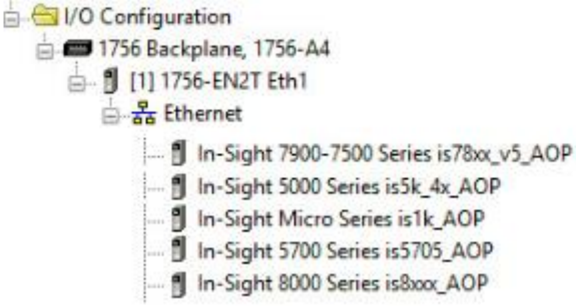
- **Requested Packet Interval (RPI):** This field specifies the Requested Packet Interval (RPI), which defines the amount of time (in milliseconds) between data exchanges across an implicit messaging connection.

> **Note**: For best results, the RPI time should be set to no more than half the time between when the In-Sight vision system completes the job execution and when the PLC requires the data.

- **Major Fault On Controller If Connection Fails While in Run Mode**: This option will cause the controller to generate a major fault when the connection fails.
- **Inhibit Module**: Checking this box prevents the PLC from attempting to establish a connection with an In-Sight vision system.

> **Note**: If the In-Sight vision system appears to be stuck in "Standby" mode, then inhibiting and re-enabling the module can cause the module to become operational again.

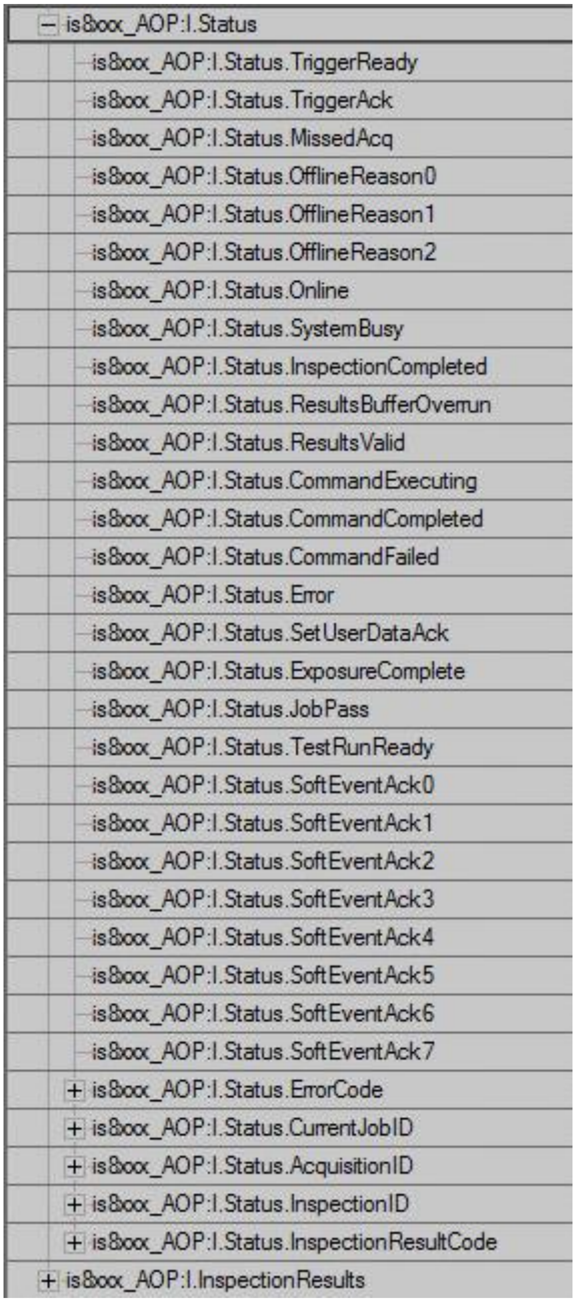
6) After adding the module to ControlLogix, the I/O tree should appear as follows:



- I/O Configuration
  - 1756 Backplane, 1756-A4
    - [1] 1756-EN2T Eth1
      - Ethernet
        - In-Sight 7900-7500 Series is78xx_v5_AOP
        - In-Sight 5000 Series is5k_4x_AOP
        - In-Sight Micro Series is1k_AOP
        - In-Sight 5700 Series is5705_AOP
        - In-Sight 8000 Series is8xxx_AOP

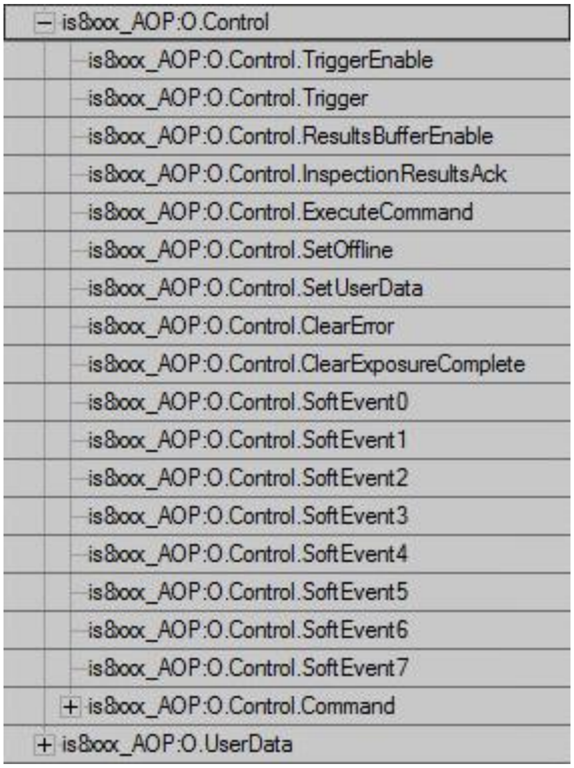
7) RSLogix 5000 will then create tags that map to the In-Sight vision system's Input and Output Data, based on the name given to the device.

- **InSight_Top:I Scheduled Input**: The table below represents the data that is sent *FROM* the In-Sight vision system to the PLC:



- is8xxx_AOP:I.Status
  - is8xxx_AOP:I.Status.TriggerReady
  - is8xxx_AOP:I.Status.TriggerAck
  - is8xxx_AOP:I.Status.MissedAcq
  - is8xxx_AOP:I.Status.OfflineReason0
  - is8xxx_AOP:I.Status.OfflineReason1
  - is8xxx_AOP:I.Status.OfflineReason2
  - is8xxx_AOP:I.Status.Online
  - is8xxx_AOP:I.Status.SystemBusy
  - is8xxx_AOP:I.Status.InspectionCompleted
  - is8xxx_AOP:I.Status.ResultsBufferOverrun
  - is8xxx_AOP:I.Status.ResultsValid
  - is8xxx_AOP:I.Status.CommandExecuting
  - is8xxx_AOP:I.Status.CommandCompleted
  - is8xxx_AOP:I.Status.CommandFailed
  - is8xxx_AOP:I.Status.Error
  - is8xxx_AOP:I.Status.SetUserDataAck
  - is8xxx_AOP:I.Status.ExposureComplete
  - is8xxx_AOP:I.Status.JobPass
  - is8xxx_AOP:I.Status.TestRunReady
  - is8xxx_AOP:I.Status.SoftEventAck0
  - is8xxx_AOP:I.Status.SoftEventAck1
  - is8xxx_AOP:I.Status.SoftEventAck2
  - is8xxx_AOP:I.Status.SoftEventAck3
  - is8xxx_AOP:I.Status.SoftEventAck4
  - is8xxx_AOP:I.Status.SoftEventAck5
  - is8xxx_AOP:I.Status.SoftEventAck6
  - is8xxx_AOP:I.Status.SoftEventAck7
  - is8xxx_AOP:I.Status.ErrorCode
  - is8xxx_AOP:I.Status.CurrentJobID
  - is8xxx_AOP:I.Status.AcquisitionID
  - is8xxx_AOP:I.Status.InspectionID
  - is8xxx_AOP:I.Status.InspectionResultCode
- is8xxx_AOP:I.InspectionResults

- **InSight_Top:O Scheduled Output**: The table below represents the data sent *TO* the In-Sight vision system from the EtherNet/IP client (ControlLogix):



- is8xxx_AOP:O.Control
  - is8xxx_AOP:O.Control.TriggerEnable
  - is8xxx_AOP:O.Control.Trigger
  - is8xxx_AOP:O.Control.ResultsBufferEnable
  - is8xxx_AOP:O.Control.InspectionResultsAck
  - is8xxx_AOP:O.Control.ExecuteCommand
  - is8xxx_AOP:O.Control.SetOffline
  - is8xxx_AOP:O.Control.SetUserData
  - is8xxx_AOP:O.Control.ClearError
  - is8xxx_AOP:O.Control.ClearExposureComplete
  - is8xxx_AOP:O.Control.SoftEvent0
  - is8xxx_AOP:O.Control.SoftEvent1
  - is8xxx_AOP:O.Control.SoftEvent2
  - is8xxx_AOP:O.Control.SoftEvent3
  - is8xxx_AOP:O.Control.SoftEvent4
  - is8xxx_AOP:O.Control.SoftEvent5
  - is8xxx_AOP:O.Control.SoftEvent6
  - is8xxx_AOP:O.Control.SoftEvent7
  - is8xxx_AOP:O.Control.Command
- is8xxx_AOP:O.UserData

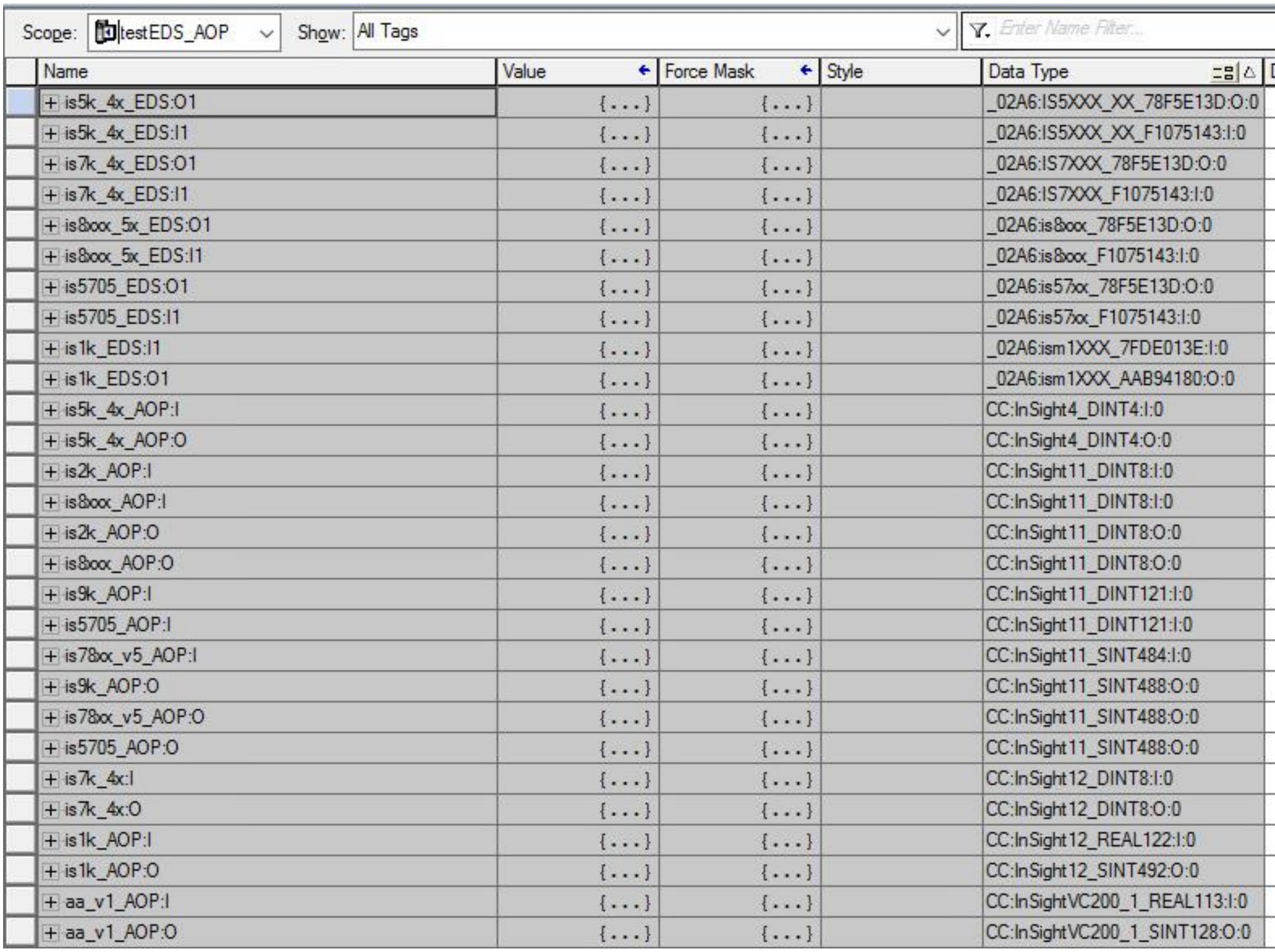
How to update a previously installed EDS profile, with the new AOP profile

These steps assume a vision system has already been setup in your RSLogix project with the EDS file.

> **Notes**:
>
> - When using Rockwell RSLogix Studio 5000 version 14 and earlier, an Allen-Bradley Generic ETHERNET-MODULE must be used. The EDS generated profile must also be used since the new AOP (1.28) requires RSLogix 15 or greater. Reference the charts in the "Integration with RSLogix" section of this document for install guidance.

1) Install AOP 1.28
    a) Make note of the EDS generated profile camera names & IP addresses that you want to replace with the AOP camera module.
2) Delete the EDS cameras from the module list (you will get tag errors)
    a) This will remove the controller tags but it will leave the tags in the ladder as Undefined tag, and the rungs will be marked with a red X or e (for error).
3) Add the cameras back using the AOP Module and use the same names and IP addresses of the deleted cameras.
    a) The icon will be the grey Rockwell Automation module
4) Delete the copy rungs needed for the EDS generated cameras
    a) These are no longer needed since the AOP creates module defined types and controller tags as structured data under the Controller->Controller tags. These rungs were needed to move the unstructured data under the MainProgram, to structured controller tags.
    b) With the AOP, when the data is sent from In-Sight to the PLC, this data will appear in the input controller tags for the In-Sight: ***In-Sight Name:I.*** And when data is sent from the PLC to In-Sight, the data should be placed in the output controller tags for the In-Sight: ***In-Sight Name****: O*
5) Search and replace all the cameras tags to update them to the new tag names.
    a) Additionally, a search and replace of the structured tags that the EDS data was copied into by the rungs. These references should be replaced with the new camera names: **In-Sight Name:I** and **In-Sight Name**: O .

       The EDS installation ladder would have used the insight11_in and insight11_out tags that were created from the rung import. These tags use the user-defined types that map out the signals for control and status. Since these rungs are no longer needed, search for those tags and replace with the **In-Sight Name:I** and **In-Sight Name:O**. The status and control parts of the tag and signal names are there already.
    b) Below is an example of systems setup with the EDS profile and the AOP profile. Note the systems setup with the EDS profile have a 1 appended at the end of each tag. While the systems setup with the AOP do not.



Scope: testEDS_AOP Show: All Tags

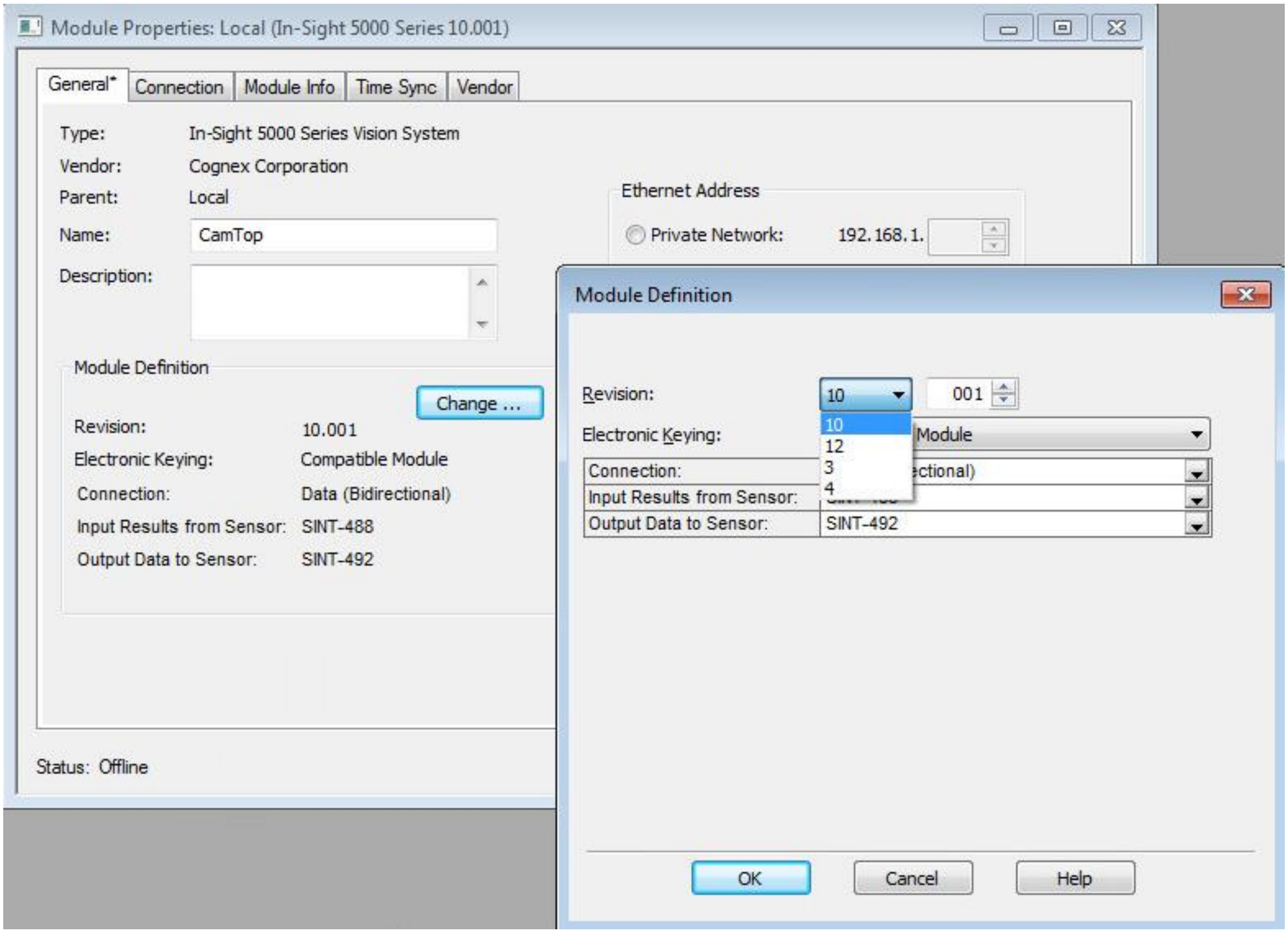
Name	Value	Force Mask	Style	Data Type
is5k_4x_EDS:O1	{...}	{...}		_02A6:IS5XXX_XX_78F5E13D:O:0
is5k_4x_EDS:I1	{...}	{...}		_02A6:IS5XXX_XX_F1075143:I:0
is7k_4x_EDS:O1	{...}	{...}		_02A6:IS7XXX_78F5E13D:O:0
is7k_4x_EDS:I1	{...}	{...}		_02A6:IS7XXX_F1075143:I:0
is8xxx_5x_EDS:O1	{...}	{...}		_02A6:is8xxx_78F5E13D:O:0
is8xxx_5x_EDS:I1	{...}	{...}		_02A6:is8xxx_F1075143:I:0
is5705_EDS:O1	{...}	{...}		_02A6:is57xx_78F5E13D:O:0
is5705_EDS:I1	{...}	{...}		_02A6:is57xx_F1075143:I:0
is1k_EDS:I1	{...}	{...}		_02A6:ism1XXX_7FDE013E:I:0
is1k_EDS:O1	{...}	{...}		_02A6:ism1XXX_AAB94180:O:0
is5k_4x_AOP:I	{...}	{...}		CC:InSight4_DINT4:I:0
is5k_4x_AOP:O	{...}	{...}		CC:InSight4_DINT4:O:0
is2k_AOP:I	{...}	{...}		CC:InSight11_DINT8:I:0
is8xxx_AOP:I	{...}	{...}		CC:InSight11_DINT8:I:0
is2k_AOP:O	{...}	{...}		CC:InSight11_DINT8:O:0
is8xxx_AOP:O	{...}	{...}		CC:InSight11_DINT8:O:0
is9k_AOP:I	{...}	{...}		CC:InSight11_DINT121:I:0
is5705_AOP:I	{...}	{...}		CC:InSight11_DINT121:I:0
is78xx_v5_AOP:I	{...}	{...}		CC:InSight11_SINT484:I:0
is9k_AOP:O	{...}	{...}		CC:InSight11_SINT488:O:0
is78xx_v5_AOP:O	{...}	{...}		CC:InSight11_SINT488:O:0
is5705_AOP:O	{...}	{...}		CC:InSight11_SINT488:O:0
is7k_4x:I	{...}	{...}		CC:InSight12_DINT8:I:0
is7k_4x:O	{...}	{...}		CC:InSight12_DINT8:O:0
is1k_AOP:I	{...}	{...}		CC:InSight12_REAL122:I:0
is1k_AOP:O	{...}	{...}		CC:InSight12_SINT492:O:0
aa_v1_AOP:I	{...}	{...}		CC:InSightVC200_1_REAL113:I:0
aa_v1_AOP:O	{...}	{...}		CC:InSightVC200_1_SINT128:O:0

How to update a previously AOP (rev 10) revision to current revision (rev 12)

This section assumes AOP 1.26 has been used. This installs revision 10. To use the new AOP version 1.28 (installs revision 12), follow the steps below. Also make note of the assembly IO changes between revisions. These are highlighted in the **Assembly I/O Changes** section of this document.

The following steps can be taken to use the new AOP:

1) Install the AOP version 1.28.
2) Open the camera module properties and press the Change…. button.



Module Properties: Local (In-Sight 5000 Series 10.001)

Type: In-Sight 5000 Series Vision System
Vendor: Cognex Corporation
Parent: Local
Name: CamTop
Description:

Ethernet Address
○ Private Network: 192.168.1.

Module Definition

Revision: 10.001
Electronic Keying: Compatible Module
Connection: Data (Bidirectional)
Input Results from Sensor: SINT-488
Output Data to Sensor: SINT-492

Status: Offline

Module Definition

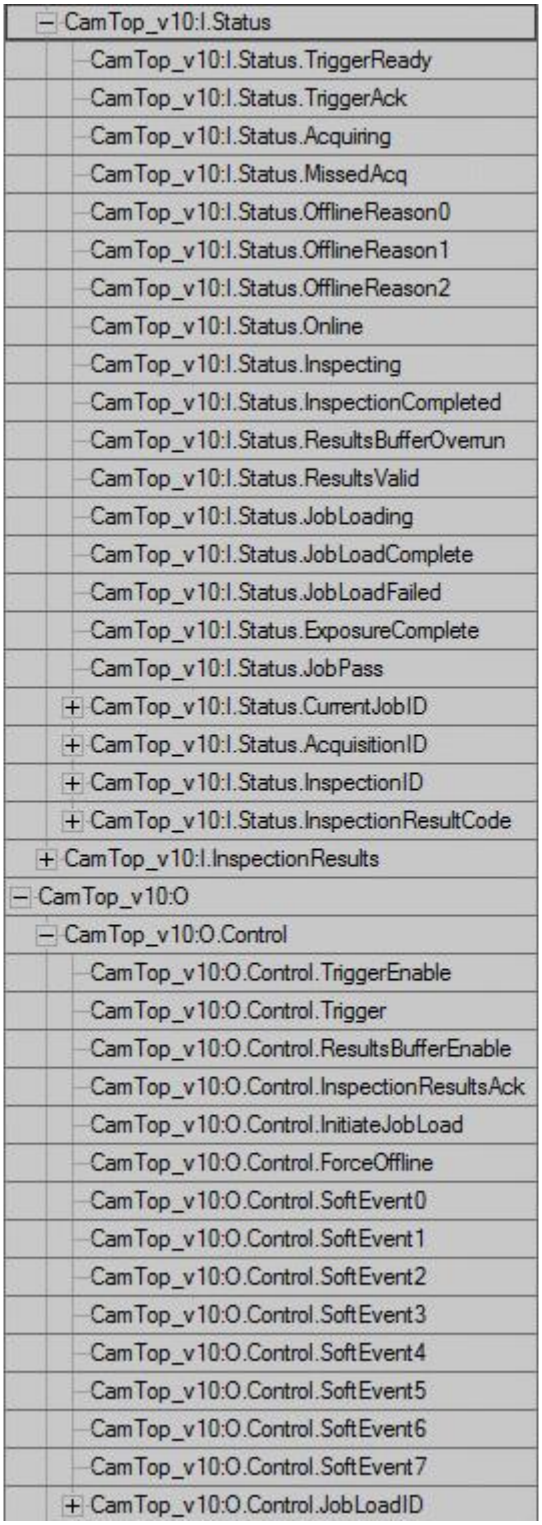
Revision: 10 001
Electronic Keying: Module
Connection: ctional)
Input Results from Sensor:
Output Data to Sensor: SINT-492

3) Select the current AOP revision (revision 12 for camera models running 4.x.x).
4) Correct any errors in tags.
    a. There are updates to the IO assembly and you may see errors in the rungs using the tags highlighted in red below.

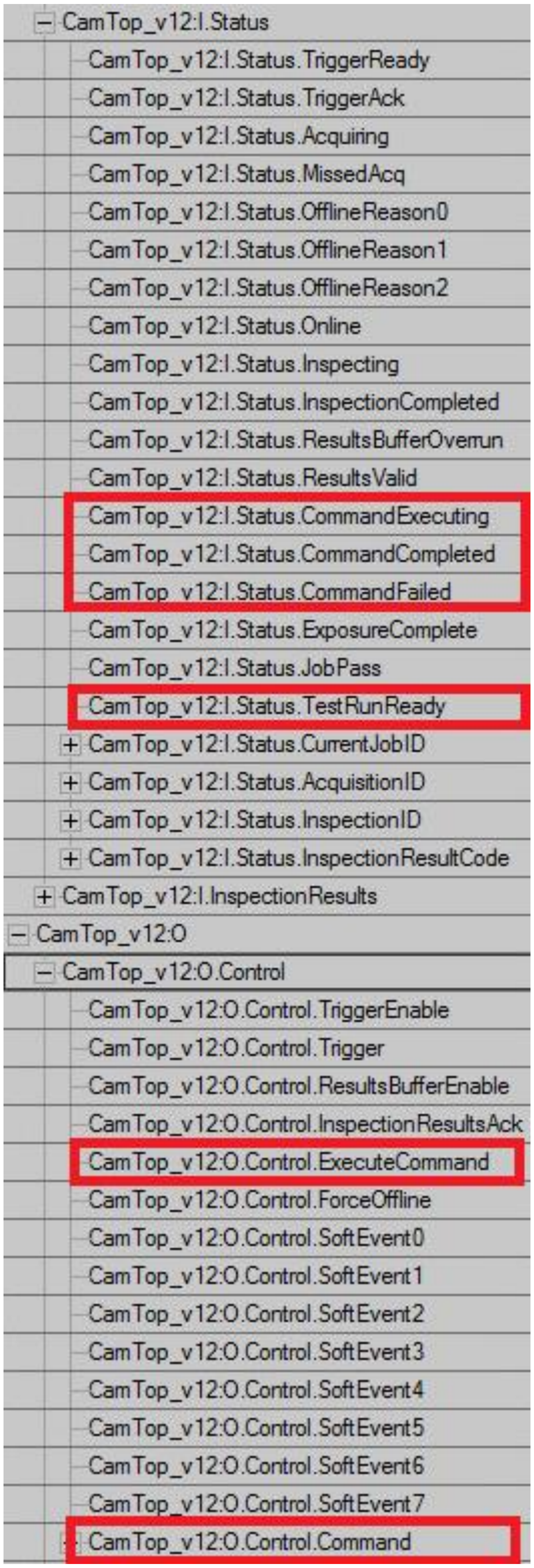
        The camera input/output tags for version 10 are on the left.

        The camera input/output tags for version 12 are on the right.

        Tag name changes are highlighted in red boxes on revision 12.



− CamTop_v10:I.Status
CamTop_v10:I.Status.TriggerReady
CamTop_v10:I.Status.TriggerAck
CamTop_v10:I.Status.Acquiring
CamTop_v10:I.Status.MissedAcq
CamTop_v10:I.Status.OfflineReason0
CamTop_v10:I.Status.OfflineReason1
CamTop_v10:I.Status.OfflineReason2
CamTop_v10:I.Status.Online
CamTop_v10:I.Status.Inspecting
CamTop_v10:I.Status.InspectionCompleted
CamTop_v10:I.Status.ResultsBufferOverrun
CamTop_v10:I.Status.ResultsValid
CamTop_v10:I.Status.JobLoading
CamTop_v10:I.Status.JobLoadComplete
CamTop_v10:I.Status.JobLoadFailed
CamTop_v10:I.Status.ExposureComplete
CamTop_v10:I.Status.JobPass
+ CamTop_v10:I.Status.CurrentJobID
+ CamTop_v10:I.Status.AcquisitionID
+ CamTop_v10:I.Status.InspectionID
+ CamTop_v10:I.Status.InspectionResultCode
+ CamTop_v10:I.InspectionResults
− CamTop_v10:O
− CamTop_v10:O.Control
CamTop_v10:O.Control.TriggerEnable
CamTop_v10:O.Control.Trigger
CamTop_v10:O.Control.ResultsBufferEnable
CamTop_v10:O.Control.InspectionResultsAck
CamTop_v10:O.Control.InitiateJobLoad
CamTop_v10:O.Control.ForceOffline
CamTop_v10:O.Control.SoftEvent0
CamTop_v10:O.Control.SoftEvent1
CamTop_v10:O.Control.SoftEvent2
CamTop_v10:O.Control.SoftEvent3
CamTop_v10:O.Control.SoftEvent4
CamTop_v10:O.Control.SoftEvent5
CamTop_v10:O.Control.SoftEvent6
CamTop_v10:O.Control.SoftEvent7
+ CamTop_v10:O.Control.JobLoadID



− CamTop_v12:I.Status
CamTop_v12:I.Status.TriggerReady
CamTop_v12:I.Status.TriggerAck
CamTop_v12:I.Status.Acquiring
CamTop_v12:I.Status.MissedAcq
CamTop_v12:I.Status.OfflineReason0
CamTop_v12:I.Status.OfflineReason1
CamTop_v12:I.Status.OfflineReason2
CamTop_v12:I.Status.Online
CamTop_v12:I.Status.Inspecting
CamTop_v12:I.Status.InspectionCompleted
CamTop_v12:I.Status.ResultsBufferOverrun
CamTop_v12:I.Status.ResultsValid
CamTop_v12:I.Status.CommandExecuting
CamTop_v12:I.Status.CommandCompleted
CamTop_v12:I.Status.CommandFailed
CamTop_v12:I.Status.ExposureComplete
CamTop_v12:I.Status.JobPass
CamTop_v12:I.Status.TestRunReady
+ CamTop_v12:I.Status.CurrentJobID
+ CamTop_v12:I.Status.AcquisitionID
+ CamTop_v12:I.Status.InspectionID
+ CamTop_v12:I.Status.InspectionResultCode
+ CamTop_v12:I.InspectionResults
− CamTop_v12:O
− CamTop_v12:O.Control
CamTop_v12:O.Control.TriggerEnable
CamTop_v12:O.Control.Trigger
CamTop_v12:O.Control.ResultsBufferEnable
CamTop_v12:O.Control.InspectionResultsAck
CamTop_v12:O.Control.ExecuteCommand
CamTop_v12:O.Control.ForceOffline
CamTop_v12:O.Control.SoftEvent0
CamTop_v12:O.Control.SoftEvent1
CamTop_v12:O.Control.SoftEvent2
CamTop_v12:O.Control.SoftEvent3
CamTop_v12:O.Control.SoftEvent4
CamTop_v12:O.Control.SoftEvent5
CamTop_v12:O.Control.SoftEvent6
CamTop_v12:O.Control.SoftEvent7
CamTop_v12:O.Control.Command

Get Data from an In-Sight Vision System – 5.x.x Spreadsheet

In order to get data from the In-Sight Explorer Spreadsheet to a ControlLogix PLC, the data must be pushed into the EtherNet/IP stack by using the WriteResultsBuffer function. This function takes a buffer of data created by the FormatOutputBuffer function and writes it to the data area in the In-Sight vision system's EtherNet/IP Input Assembly (input to the network). This data is then transferred to the PLC during the next RPI cycle.

The EtherNet/IP buffer writes to this data area are queued, meaning that they do not happen during Spreadsheet execution, but as a separate event slightly after the execution of the Spreadsheet. This means that there is a small window of time during which an inspection will be complete, but a new buffer write will not have resolved.

It is important to use the *ResultsValid* bit to decide when to read data to prevent subtle intermittent timing issues in your PLC logic that can arise from this setup, as *ResultsValid* will only become high when data are ready to read, and not before.

- EXAMPLE: Getting 32 bit integer data from an In-Sight vision system
- EXAMPLE: Getting floating point value data from an In-Sight vision system

---

EXAMPLE: Getting 32 bit integer data from an In-Sight vision system

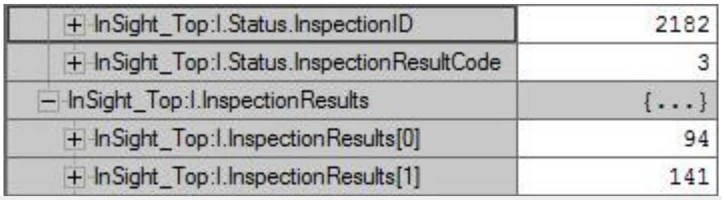
The following steps explain how to format the data that will be sent from an In-Sight vision system to a ControlLogix PLC.

1. To begin, using In-Sight Explorer, create a new job.
2. From the Palette's Snippets tab, add these two Snippets to the spreadsheet: *Acquisition* > *AcqCounter* and *Math & Logic* > *Random*.
3. Open the AcquireImage cell and set the **Trigger** parameter to *Continuous*.
4. Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category and double-click on the FormatOutputBuffer function to insert it into the spreadsheet.
5. From the FormatOutputBuffer dialog, click on the *Add* button. This will initiate the cell selection mode; select the "Scaled random number" cell of the *Random* snippet.
6. From the FormatOutputBuffer dialog, use the *Data Type* pull-down menu to change the *Data Type* to *32 bit integer*.
7. From the FormatOutputBuffer dialog, click on the *Add* button again. This will initiate cell selection mode; select the count cell of the *AcqCounter* snippet.
8. Close the FormatOutputBuffer by clicking the *OK* button.
9. Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category, click on the *Network* subcategory and double-click on the WriteResultsBuffer function to insert it into the spreadsheet.

10. Set the WriteResultsBuffer function's *Protocol* parameter to *EtherNet/IP*, and the *Buffer* parameter as a cell reference to the just created FormatOutputBuffer Buffer data structure. Set a value of 3 in the WriteResultsBuffer Results Code parameter. (Or reference any cell with integer data)

11. Place the In-Sight vision system Online.

12. The data should now be displayed in the *Controller* -> *Controller Tags* node of RSLogix 5000.



+ InSight_Top:I.Status.InspectionID	2182
+ InSight_Top:I.Status.InspectionResultCode	3
− InSight_Top:I.InspectionResults	{...}
+ InSight_Top:I.InspectionResults[0]	94
+ InSight_Top:I.InspectionResults[1]	141

> **Note**: The RSLogix Monitor Tags tab must be selected in order to display the data values.

EXAMPLE: Getting floating point value data from an In-Sight vision system

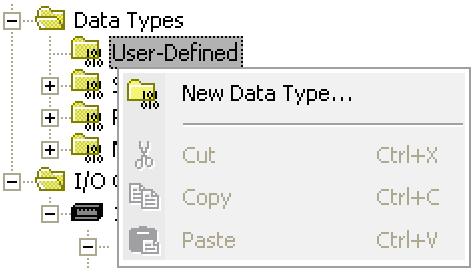
If a floating point or a mix of data types needs to be sent, in ControlLogix create a User Defined Type. Copy the floating point data into the newly created User Defined Type from the In-Sight vision system's input tag.

The steps below illustrate how to copy two cells as floating point values:

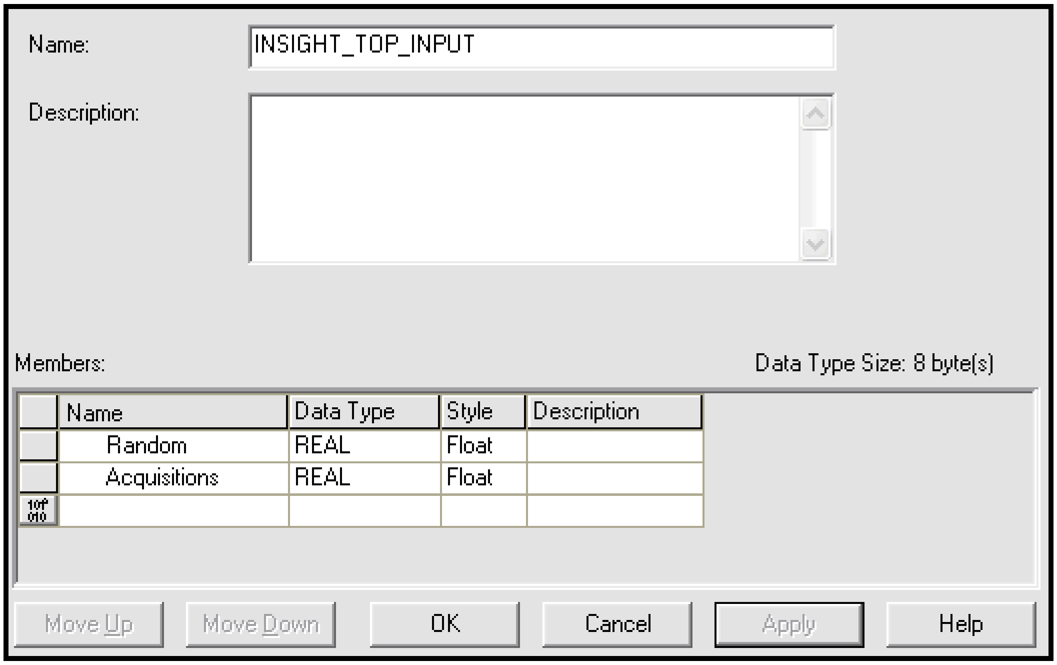
1. Place the In-Sight vision system Offline.
2. Open the previously created FormatOutputBuffer function.
3. From the FormatOutputBuffer dialog, use the *Data Type* pull-down menu to change the *Data Type* from *32 bit integer* to *32 bit floating point*.

   > **Note**: Matching the *Data Type* being used by the vision system is not necessarily required. The bits are copied verbatim from the vision system's output buffers in the order that they appear in FormatOutputBuffer or the Format Outputs tab, regardless of the data type used to receive the buffer in the PLC.

4. Place the In-Sight vision system back Online.
5. In RSLogix, go into Offline mode.
6. Within RSLogix, right-click on the *Data Types* -> *User-Defined* folder and select *New Data Type* from the menu.



7. Give the new data type a descriptive name and add two floating point values to the *Members:* list.



Name: INSIGHT_TOP_INPUT

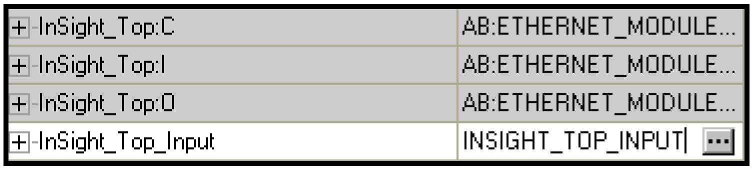
Description:

Members:

Data Type Size: 8 byte(s)

Name	Data Type	Style	Description
Random	REAL	Float	
Acquisitions	REAL	Float	

8. Next, add a new tag to the project's Controller Tags, using the user defined data type.

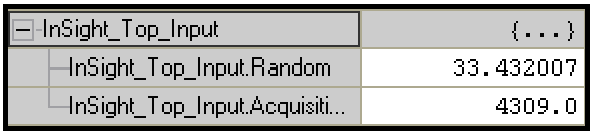


InSight_Top:C	AB:ETHERNET_MODULE...
InSight_Top:I	AB:ETHERNET_MODULE...
InSight_Top:O	AB:ETHERNET_MODULE...
InSight_Top_Input	INSIGHT_TOP_INPUT

9. Finally, add a COP instruction to the MainRoutine ladder logic program, which will copy the data from the **InSight_Top:I.InspectionResults[0]** tag to the **InSight_Top_Input** tag.

> **Note**: The Length field is the number of the input structures that should be copied to the output. In this case, the input structures are 32-bit floating point values, so the COP command copies 64 bits from the vision system's output buffer into the user-defined type in the PLC. If we were using 16-bit integer types (INT)in the PLC's receiving buffers, then we would have used a length of 4 to get the same 64 bits, and if we were using 8-bit integer types (SINT), then we would have used a length of 8 to copy all 64 bits.

10. After this program has been downloaded to the ControlLogix PLC, the data from the Random and Acquisition Counter In-Sight Explorer spreadsheet cells will now appear in the **InSight_Top_Input** tag in the floating point format:



InSight_Top_Input	{...}
InSight_Top_Input.Random	33.432007
InSight_Top_Input.Acquisiti...	4309.0

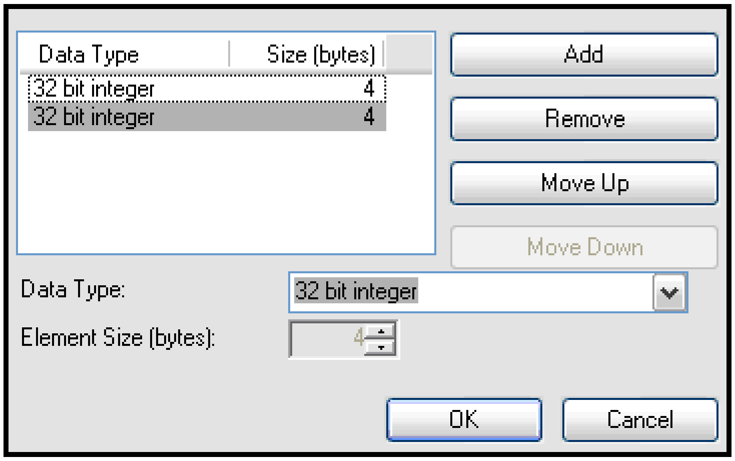
Send Data To an In-Sight Vision System – 5.x.x Spreadsheet

In order to send data from the ControlLogix PLC to the In-Sight Explorer spreadsheet, the data must be pulled from the EtherNet/IP stack by using the ReadUserDataBuffer function. This function takes the data format created within the FormatInputBuffer function and reads the data from the data area of the EtherNet/IP *Output Assembly* (output from the network), and formats this data into the In-Sight Explorer Spreadsheet. This data is received from the PLC every RPI cycle; the corresponding Spreadsheet data is updated only when the Spreadsheet executes.

EXAMPLE: Sending 32 bit integer data to an In-Sight Vision System

For this example, create a new job within In-Sight Explorer, then perform the following steps to configure the data that will be received by the ControlLogix PLC.

1) Open the AcquireImage cell and set the **Trigger** parameter to *Continuous*.
2) Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category and double-click on the FormatInputBuffer function to insert it into the spreadsheet.
3) From the FormatInputBuffer dialog, click on the **Add** button and add two 32-bit integers to the list.



Data Type	Size (bytes)
32 bit integer	4
32 bit integer	4

Data Type: 32 bit integer
Element Size (bytes): 4

4) Close the FormatInputBuffer by clicking the *OK* button.

5) Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category, click on the *Network* subcategory and double-click on the ReadUserDataBuffer function to insert it into the Spreadsheet.

6) Set the ReadUserDataBuffer function's *Protocol* parameter to *EtherNet/IP*, and the *Buffer* parameter as a cell reference to the just created FormatOutputBuffer Buffer data structure.
7) The Data Access functions will automatically be added to the spreadsheet based on the fields added to the FormatOutputBuffer function.

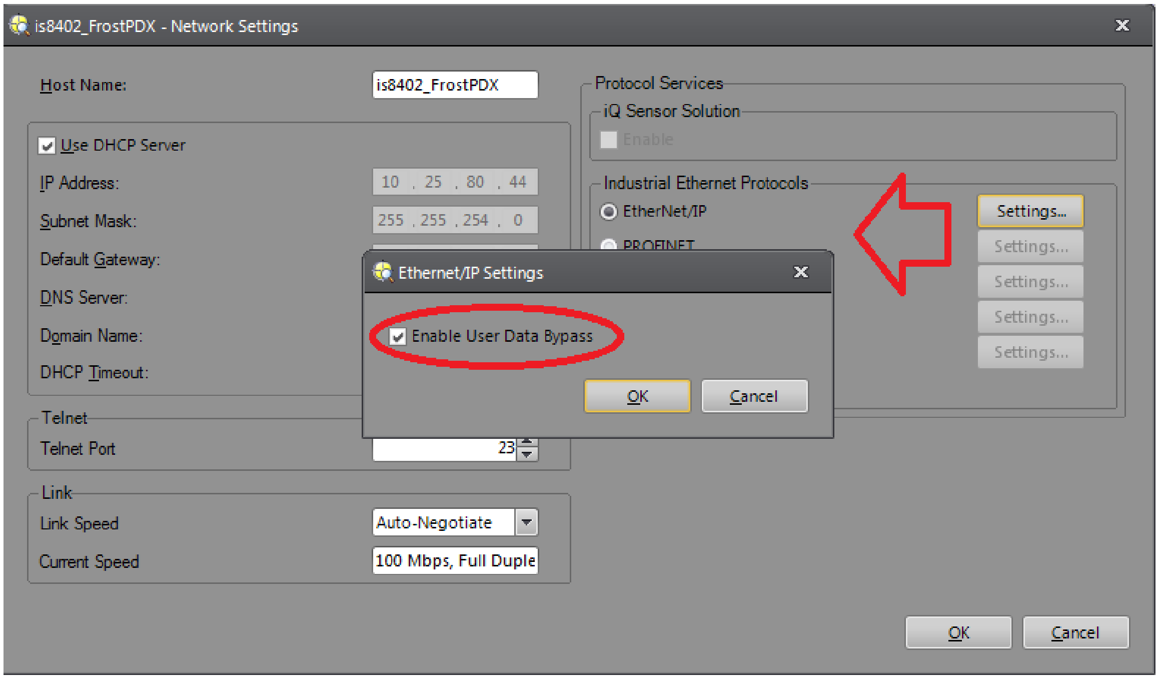
8) Place the In-Sight vision system in Online mode.

9) Within RSLogix 5000, open the *Controller Tags* dialog and change the value of the **InSight_Top:O.UserData[0]** and **InSight_Top:O.UserData[1]** tags.

10) Determine if the User Data handshaking is needed. If the User Data Bypass is enabled, the exchange of data will mimic that of the 4.xx cameras. If the User Data Bypass is not enabled (default state), this new bit will add additional handshaking to

the data exchange between PLC and In-Sight. Additional steps will be needed to be done to display the data in the spreadsheet.

a) If the Enable User Data Bypass is enabled in the Network Setting Dialog for the camera:



is8402_FrostPDX - Network Settings

Host Name: is8402_FrostPDX

☑ Use DHCP Server

IP Address: 10 . 25 . 80 . 44

Subnet Mask: 255 . 255 . 254 . 0

Default Gateway:

DNS Server:

Domain Name:

DHCP Timeout:

Telnet

Telnet Port 23

Link

Link Speed Auto-Negotiate

Current Speed 100 Mbps, Full Duple

Protocol Services

iQ Sensor Solution

☐ Enable

Industrial Ethernet Protocols

● EtherNet/IP

○ PROFINET

Ethernet/IP Settings

☑ Enable User Data Bypass

The values in In-Sight Explorer spreadsheet should change immediately.

b) If the Enable User Data Bypass is <u>**NOT**</u> enabled in the Network Setting Dialog for the camera (default), then:

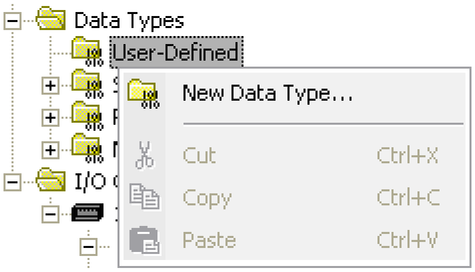
  i) Set the User Data tag, SetUserData bit is a named tag, for this example it would be InSight_Top:O.Control.SetUserData.

  ii) Clear the User Data tag above once you receive the User Data Ack bit. SetUserDataAck bit is a named tag, for this example it would be InSIght_Top:I.Status. SetUserDataAck.

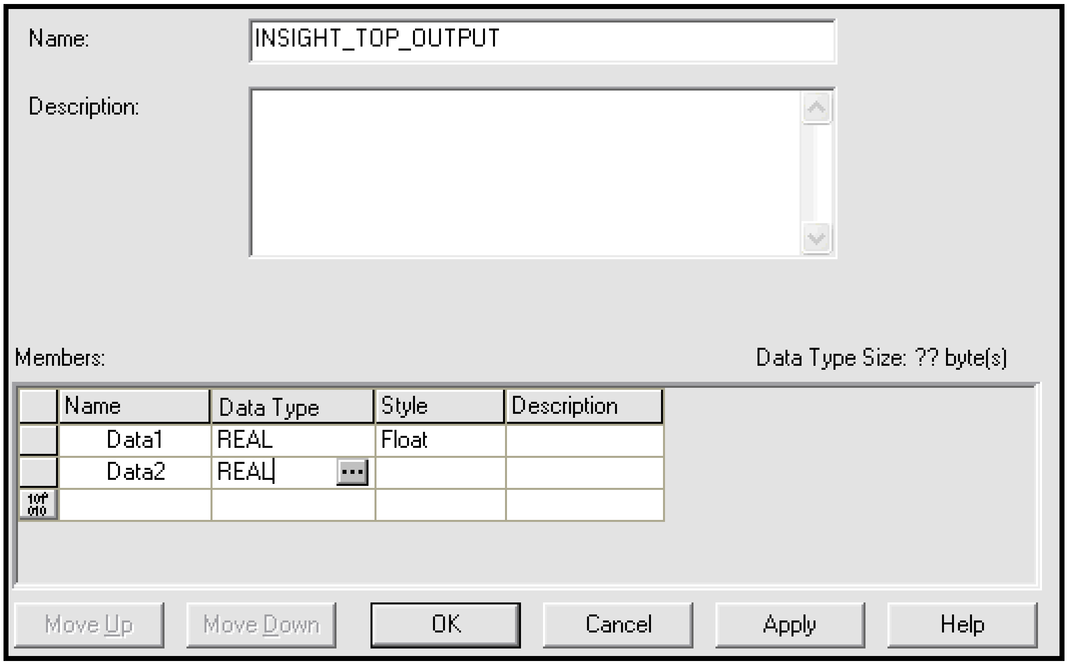
EXAMPLE: Sending floating point value data to an In-Sight vision system

If another data type, for instance a type other than the 32-bit integers type, or a mix of data types needs to be sent, the simplest method to accomplish this is to create a *User Defined Type* in ControlLogix, and then to copy that data from the In-Sight vision system's output tag to the newly created *User Defined Type*. The steps below illustrate how to copy to two cells as floating point values:

1) In RSLogix, go into Offline mode.
2) Within RSLogix, right-click on the *User-Defined ->Data Types* folder and select *New Data Type...* from the menu.



3) Give the new data type a descriptive name and add two floating point values to the *Members:* list.



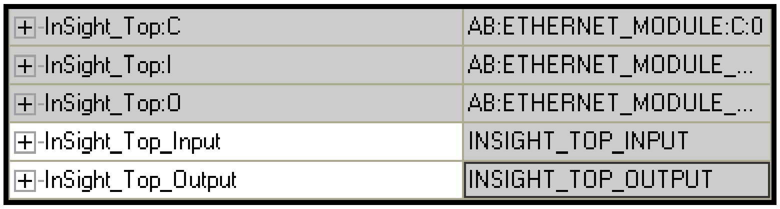
Name: INSIGHT_TOP_OUTPUT

Description:

Members: Data Type Size: ?? byte(s)

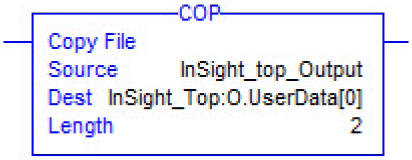
Name	Data Type	Style	Description
Data1	REAL	Float	
Data2	REAL		

4) Next, add a new tag to the project's Controller Tags, using the user defined data type, as shown in the example below.



InSight_Top:C	AB:ETHERNET_MODULE:C:0
InSight_Top:I	AB:ETHERNET_MODULE_...
InSight_Top:O	AB:ETHERNET_MODULE_...
InSight_Top_Input	INSIGHT_TOP_INPUT
InSight_Top_Output	INSIGHT_TOP_OUTPUT

5) Finally, add a COP instruction to the MainRoutine ladder logic program, which will copy the data from the **InSight_Top_Output** tag to **InSight_Top:O.UserData[0]** and **InSight_Top:O.UserData[1]** tags.



COP
Copy File
Source InSight_top_Output
Dest InSight_Top:O.UserData[0]
Length 2

6) Place the In-Sight vision system Offline.

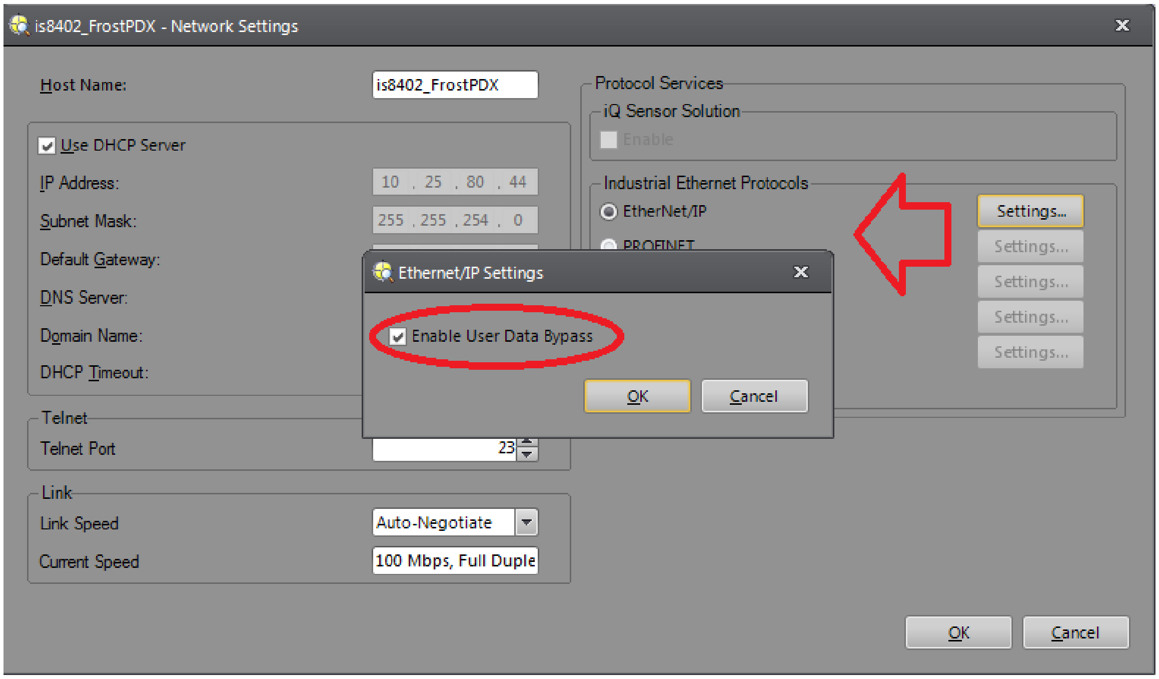
7) Open the previously created FormatInputBuffer function, and change the *Data Type* values of both to *32 bit floating point*.

8) Place the In-Sight vision system back Online.

9) After this program has been downloaded to the ControlLogix PLC, the data in the **InSight_Top_Output** tag will now appear in the floating point format.

10) Determine if the User Data handshaking is needed. If the User Data Bypass is enabled, the exchange of data will mimic that of the 4.xx cameras. If the User Data Bypass is not enabled (default state), this new bit will add additional handshaking to the data exchange between PLC and In-Sight. Additional steps will be needed to be done to display the data in the spreadsheet.

a) If the Enable User Data Bypass is enabled (checked ON) in the Network Setting Dialog for the camera, the values in In-Sight Explorer spreadsheet will update immediately.



Host Name: is8402_FrostPDX

☑ Use DHCP Server

IP Address: 10 . 25 . 80 . 44

Subnet Mask: 255 . 255 . 254 . 0

Default Gateway:

DNS Server:

Domain Name:

DHCP Timeout:

Telnet Port: 23

Link Speed: Auto-Negotiate

Current Speed: 100 Mbps, Full Duple

☑ Enable User Data Bypass

b) If the Enable User Data Bypass is **NOT** enabled (unchecked which is the default state) in the Network Setting Dialog for the camera, then:

i) Set the User Data tag, SetUserData bit is a named tag, for this example it would be InSight_Top:O.Control.SetUserData.

ii) Clear the User Data tag above once you receive the User Data Ack bit. SetUserDataAck bit is a named tag, for this example it would be InSIght_Top:I.Status. SetUserDataAck.

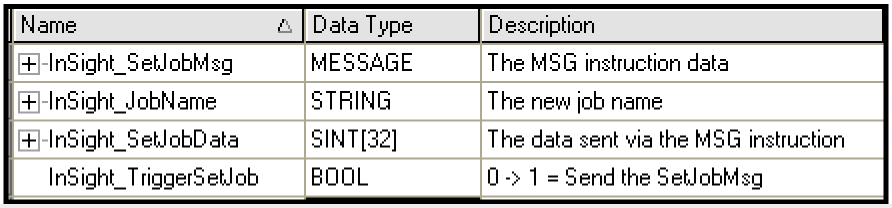
Explicit Messages - In-Sight Firmware 5.x.x

Unlike implicit messages, explicit messages are sent to a specific device and that device always responds with a reply to that message. As a result, explicit messages are better suited for operations that occur less frequently. Explicit messages can be used to read and write the Attributes in the EtherNet/IP Vision Object of the In-Sight Vision System, which may be used for changing jobs, acquiring images, sending Native Mode commands and retrieving result data.

Example: Change a Job

The most common explicit message sent to an In-Sight vision system from a ControlLogix PLC is an instruction to change a job. Explicit messages are sent from a ControlLogix PLC using MSG instructions. The following steps show how to add a MSG instruction in RSLogix to change a job on an In-Sight vision system:

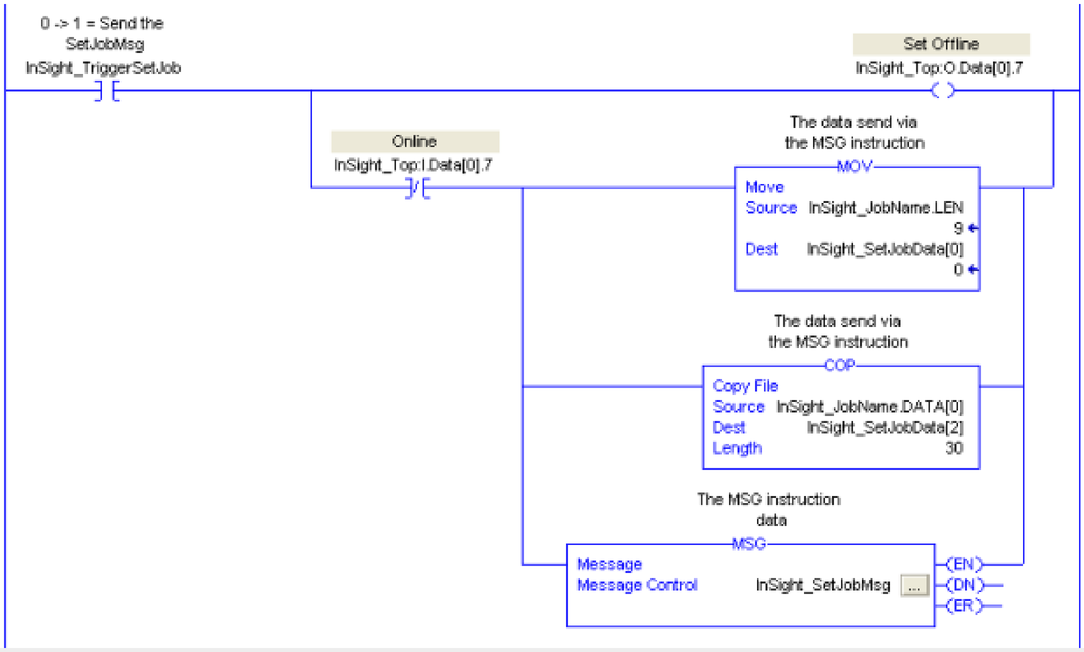
1. Add the following tags to the ControlLogix **Controller Tags** dialog:



Name △	Data Type	Description
⊞-InSight_SetJobMsg	MESSAGE	The MSG instruction data
⊞-InSight_JobName	STRING	The new job name
⊞-InSight_SetJobData	SINT[32]	The data sent via the MSG instruction
InSight_TriggerSetJob	BOOL	0 -> 1 = Send the SetJobMsg

> **Note**: This example assumes a maximum length of 30 characters for the job name. If your job's name is longer, then you will need to extend the SetJobData buffer to include a number of SINT structures equal to the number of characters in your longest job name plus two.

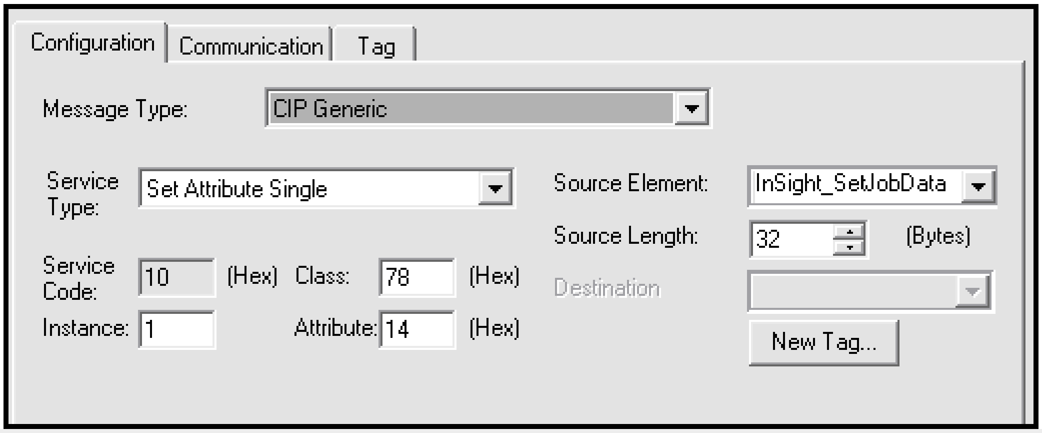
2. Create the following rung in your RSLogix 5000 project:


0 -> 1 = Send the SetJobMsg
InSight_TriggerSetJob
Set Offline
InSight_Top:O.Data[0].7
Online
InSight_Top:I.Data[0].7
The data send via the MSG instruction
MOV
Move
Source InSight_JobName.LEN
9
Dest InSight_SetJobData[0]
0
The data send via the MSG instruction
COP
Copy File
Source InSight_JobName.DATA[0]
Dest InSight_SetJobData[2]
Length 30
The MSG instruction data
MSG
Message
Message Control InSight_SetJobMsg ...
EN
DN
ER

> **Note**: This snippet is copying the length of the job name into the first two bytes of the SetJobData buffer, the string data into the same buffer starting at the third byte, and then sends that buffer to the JobName attribute of the vision system. Make sure to add the ".job" file extension onto the end of the job name.

This rung uses the *Set Offline* bit in the implicit connection to force the In-Sight vision system Offline, because the *JobName* attribute of the *Vision Object* requires the In-Sight vision system to be Offline before it will change the job. The *Set Offline* bit waits for the In-Sight vision system to bring the Online bit low, then sets up the data containing the new job name and sends the MSG instruction to the In-Sight vision system. After the job change is completed, the falling edge of the *TriggerSetJob* tag will cause the In-Sight vision system to go back Online.

3. To setup the MSG instruction, click on the *InSight_SetJobMsg …* button. This will cause the **Message Configuration** dialog to appear:



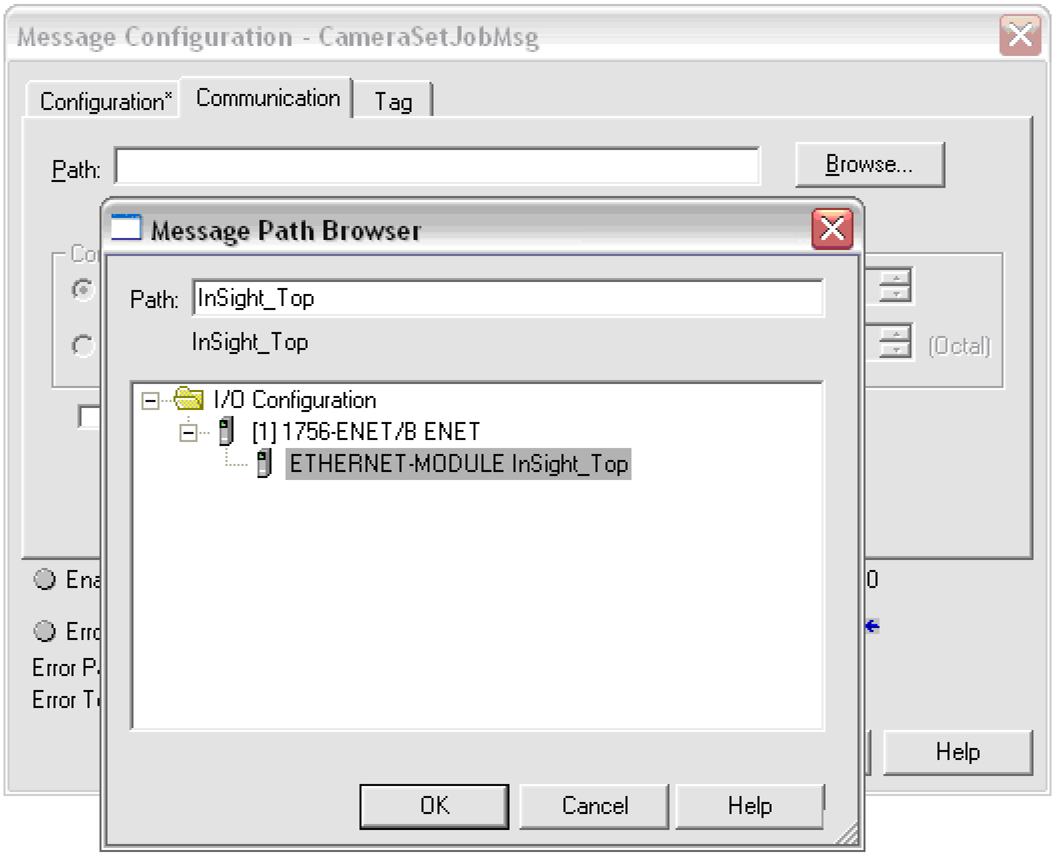
Message Type: CIP Generic
Service Type: Set Attribute Single
Service Code: 10 (Hex)
Class: 78 (Hex)
Instance: 1
Attribute: 14 (Hex)
Source Element: InSight_SetJobData
Source Length: 32 (Bytes)

> **Note**: This example assumes a maximum length of 30 characters for the job name. If your job's name is longer, then you will need to input a Source Length equal to the number of characters in your longest job name plus two.

The following fields need to be configured:

- **Message Type:** This is the type of message that will be sent; choose *CIP Generic* to send a message to vendor specific objects like the *Vision Object*.
- **Service Type:** This is the service code that will be sent to the object; select *Set Attribute Single* to change a single attribute of the *Vision Object*.
- **Service Code:** This field allows any type of service to the object; this field is disabled if a predefined service type is selected.
- **Class:** This is the identifier of the class that the message will be sent to; the ID of the *Vision Object* is *Hex 78*.
- **Instance:** This field specifies the instance number of the object that the message will be sent to; for In-Sight vision systems, the *Vision Object* only has one instance, so this field should be set to 1.
- **Attribute:** The attribute number that the message will be sent to; in this case, the *JobName* attribute has the ID of 14 Hex.
- **Source Element:** This field indicates the source for the data that is being sent with the message. For the *JobName* attribute, a *String* with a 2 byte length header needs to be sent. This string was formatted earlier using the COP and MOV instructions in the run that was created in the previous steps. This field should be set to **InSight_SetJobData** to send the new job name to the *JobName* attribute.
- **Source Length:** This field indicates the number of bytes of the source element that will be sent to the object. In this instance, because all of the data in the source element needs to be sent, the bytes should be set to 32.

4. The MSG instruction now needs to be told which device should have the explicit message sent to it. Click on the *Communication* tab of the **Message Configuration** dialog and press the *Browse...* button. Choose the In-Sight vision system from the I/O configuration as shown below:



Message Configuration - CameraSetJobMsg

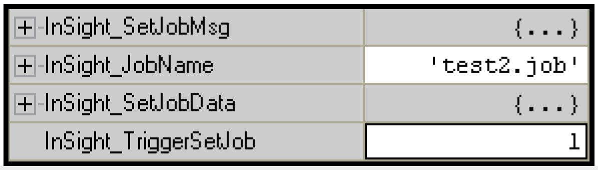
Path:

Message Path Browser

Path: InSight_Top

InSight_Top

5. After downloading the program to the ControlLogix PLC, the current job should be able to be changed with RSLogix by using the **InSight_JobName** and the **InSight_TriggerSetJob** controller tags:



⊞ InSight_SetJobMsg	{...}
⊞ InSight_JobName	'test2.job'
⊞ InSight_SetJobData	{...}
InSight_TriggerSetJob	1

> **Note**: Make sure to add the ".job" extension at the end of the job name.

6. The *Command Executing*(which can be on for a maximum of 2 seconds) and *Command Failed* bits can be used to determine if the job loaded successfully.

1. First would be to set the camera offline by setting the Offline bit high.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
22	0	Set Offline	Reserved		Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Reserved							
	2	Reserved				Clear Exposure Complete	Clear Error	Reserved	Set User Data
	3	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	4	Command							
	5								
	6 .. 7	Reserved							
	8	User Data 0							
	…								
	495	User Data 487							

2. Set the job ID into bytes 4&5 of the command block. The job name on the camera must start with a JobID number (1-999).

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
22	0	Set Offline	Reserved		Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Reserved							
	2	Reserved				Clear Exposure Complete	Clear Error	Reserved	Set User Data
	3	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	4	Command							

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	5								
	6 .. 7	Reserved							
	8	User Data 0							
	...								
	495	User Data 487							

Byte 4-5

Byte	Name	Description
4 - 5	Command	This is a 16-bit integer used to indicate the Job ID number (1-999) of the job to load when the *Execute Command* bit is set by the PLC. The Command field must be held constant between the rising edge of the *Execute Command* signal and the rising edge of the Command Completed signal, or the results will be indeterminate.

3. Set the Execute Command bit in the command block

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
22	0	Set Offline	Reserved		Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Reserved							
	2	Reserved				Clear Exposure Complete	Clear Error	Reserved	Set User Data
	3	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	4	Command							
	5								
	6 .. 7	Reserved							
	8	User Data 0							
	...								
	495	User Data 487							

Byte 0

Bit	Name	Description
7	Set Offline	When this bit is set, the In-Sight vision system is taken Offline until the bit is cleared again.
6-5	Reserved	Unused.
4	Execute Command	When set, the vision system loads the job ID specified in the Command field. The signal must be held high until the *Command Completed* signal is set. The falling edge of this signal (prior to Command Complete) is interpreted as an abort request.

4. Wait for Command Complete or Command Failed to be set before clearing the ExecuteCommand bit and the Offline bit.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
13	0	Online	Offline Reason			Missed Acq	Reserved	Trigger Ack	Trigger Ready
	1	Error	Command Failed	Command Completed	Command Executing	Results Valid	Results Buffer Overrun	Inspection Completed	System Busy
	2	Reserved	Reserved	Reserved	Job Pass	Exposure Complete	Reserved	Reserved	Set User Data Ack
	3	Soft Event Ack 7	Soft Event Ack 6	Soft Event Ack 5	Soft Event Ack 4	Soft Event Ack 3	Soft Event Ack 2	Soft Event Ack 1	Soft Event Ack 0
	4	Error Code (16-bit integer)							
	5								
	6	Reserved (16-bit integer)							
	7								
	8	Current Job ID (16-bit integer)							
	9								
	10	Acquisition ID (16-bit integer)							
	11								
	12	Inspection ID (16-bit integer)							
	13								
	14	Inspection Result Code (16-bit integer)							
	15								
	16	Inspection Results 0							
	…								
	499	Inspection Results 483							

Byte 1

Bit	Name	Description
7	Error	This bit is set when an error has occurred, which is defined in the Error Code field.
6	Command Failed	This bit is set to 1 to indicate that Job Load has failed to run to completion. It is cleared when a new job is loaded by the PLC/HMI. If the job is changed through In-Sight Explorer, this bit does not change. This bit is always set prior to setting the *Command Completed* bit.
5	Command Completed	This bit is set to indicate that Job Load has completed. When a Command completes the *Command Executing* bit goes low and if the *Execute Command* bit is still high, the *Command Completed* bit is set. If the Command did not successfully complete, the *Command Failed* bit is also set.

Get Data from an In-Sight Vision System - EasyBuilder

EtherNet/IP Communications - In-Sight 5.x.x Firmware

First setup In-Sight to communicate to the PLC:

1. In the *Communications* group box, press the **Add Device** button.
2. In the *Device Setup* group box, from the *Device* drop-down list, select *PLC/Motion Controller*.
3. From the *Manufacturer* drop-down list, select the PLC/MC manufacturer. If your PLC/MC manufacturer is not listed, choose *Other*.
4. From the *Protocol* drop-down list, select your communication protocol. The communication protocols listed are contingent upon the *Manufacturer* selection.
5. Press the **OK** button.

Next, setup the data to send / receive from the PLC. Data from EasyBuilder jobs, such as individual tool results or overall job results (output only), can be communicated over Ethernet/IP. Input data will be read after the sensor acquires an image, and output data will be sent after the sensor completes its job execution. To format your data:

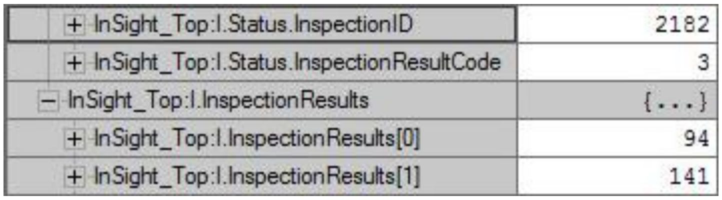
6. Click either the *Format Input Data* or *Format Output Data* tab.
7. Press the **Add** button to launch the *Select Input Data* or *Select Output Data* dialogs.
8. The *Select Input Data* or *Select Output Data* dialogs contain the data from any Location or Inspection Tools that were added to your job and the overall job results (output only). From the dialog, select the appropriate data that you want to be sent from (or to) the In-Sight sensor, and press the **OK** button.
9. After you have added your data, you can modify the default data type of the selected data by choosing a different data type in the *Data Type* drop-down list.

> **Notes**:
>
> - If you select *String* from the *Data Type* drop-down list, the *Element Size (bytes)* control will be enabled and you can specify the string length by choosing the correct number of bytes.

10. Optionally, you can manually assign a user-defined Index to the Data Type by using the Element Index control. When the Element Index control is not manually defined, the Index will be automatically assigned to each Data Type when leaving the Communication Application step or saving the job.

> **Notes**:
>
> - If an invalid Index is assigned, the job will fail. The job can also fail when the user-defined Index becomes no longer valid after changing the Data Type or moving, adding or removing the Input Data.
> - Once Index is manually assigned, the Index will no longer be assigned automatically to the Data Type. If Index is manually defined or changed and the job fails because of the Index, it must be manually corrected.

11. Rearrange the order of the data that will be sent or received by selecting the data from the list and clicking either the **Up** or **Down** buttons to set your desired order.
12. Place the In-Sight vision system Online.
13. Trigger the system.
14. The data should now be displayed in the *Controller -> Program Tags* node of RSLogix 5000.



InSight_Top:I.Status.InspectionID	2182
InSight_Top:I.Status.InspectionResultCode	3
InSight_Top:I.InspectionResults	{...}
InSight_Top:I.InspectionResults[0]	94
InSight_Top:I.InspectionResults[1]	141

Send Data to an In-Sight Vision System – EasyBuilder

EtherNet/IP Communications - In-Sight 5.x.x Firmware

First setup In-Sight to communicate to the PLC:

1) In the *Communications* group box, press the **Add Device** button.
2) In the *Device Setup* group box, from the *Device* drop-down list, select *PLC/Motion Controller*.
3) From the *Manufacturer* drop-down list, select the PLC/MC manufacturer. If your PLC/MC manufacturer is not listed, choose *Other*.
4) From the *Protocol* drop-down list, select your communication protocol. The communication protocols listed are contingent upon the *Manufacturer* selection.
5) Press the **OK** button.

Next, setup the data to send / receive from the PLC. Data from EasyBuilder jobs, such as individual tool results or overall job results (output only), can be communicated over Ethernet/IP. Input data will be read after the sensor acquires an image, and output data will be sent after the sensor completes its job execution. To format your data:

6) Click either the *Format Input Data* or *Format Output Data* tab.
7) Press the **Add** button to launch the *Select Input Data* or *Select Output Data* dialogs.
8) The *Select Input Data* or *Select Output Data* dialogs contain the data from any Location or Inspection Tools that were added to your job and the overall job results (output only). From the dialog, select the appropriate data that you want to be sent from (or to) the In-Sight sensor, and press the **OK** button.
9) After you have added your data, you can modify the default data type of the selected data by choosing a different data type in the *Data Type* drop-down list.

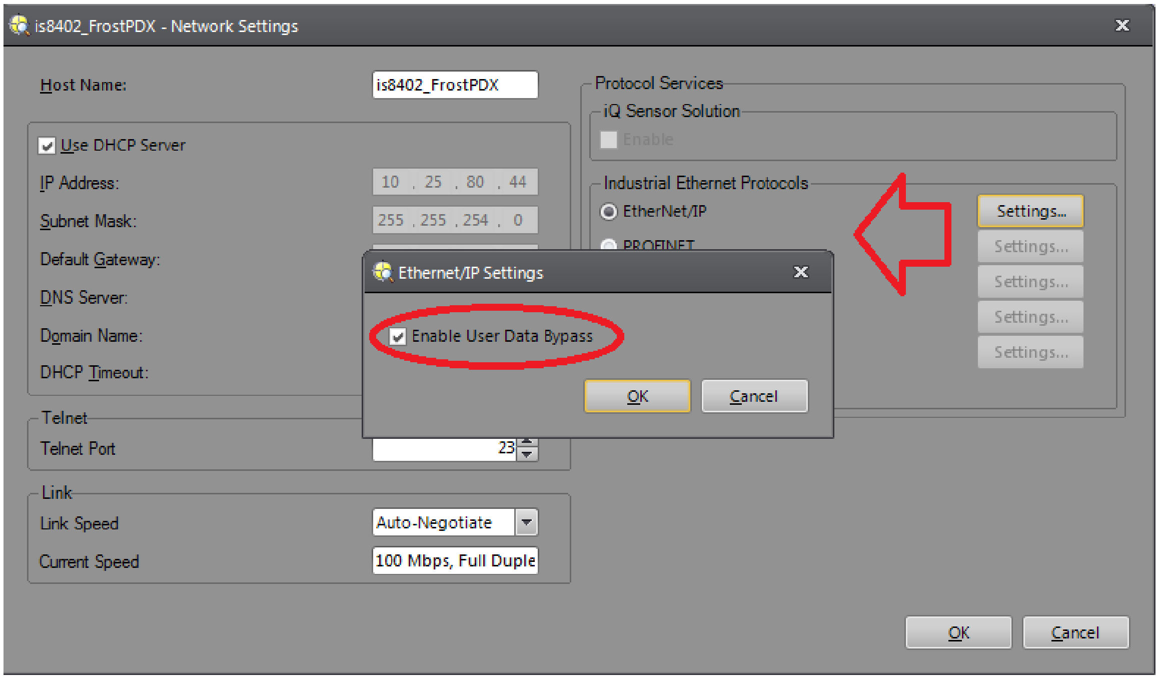
> **Notes**:
>
> - If you select *String* from the *Data Type* drop-down list, the *Element Size (bytes)* control will be enabled and you can specify the string length by choosing the correct number of bytes.

10) Optionally, you can manually assign a user-defined Index to the Data Type by using the Element Index control. When the Element Index control is not manually defined, the Index will be automatically assigned to each Data Type when leaving the Communication Application step or saving the job.

> **Notes**:
>
> - If an invalid Index is assigned, the job will fail. The job can also fail when the user-defined Index becomes no longer valid after changing the Data Type or moving, adding or removing the Input Data.
> - Once Index is manually assigned, the Index will no longer be assigned automatically to the Data Type. If Index is manually defined or changed and the job fails because of the Index, it must be manually corrected.

11) Rearrange the order of the data that will be sent or received by selecting the data from the list and clicking either the **Up** or **Down** buttons to set your desired order.
12) Place the In-Sight vision system in Online mode.
13) Within RSLogix 5000, open the *Program Tags* dialog and change the value of the **InSight_Top:O.Data[1]** and **InSight_Top:O.Data[2]** tags.
14) Determine if the User Data handshaking is needed. If the User Data Bypass is enabled, the exchange of data will mimic that of the 4.xx cameras. If the User Data Bypass is not enabled (default state), this new bit will add additional handshaking to the data exchange between PLC and In-Sight. Additional steps will be needed to be done to display the data in the spreadsheet.

a) If the Enable User Data Bypass is enabled (checked ON) in the Network Setting Dialog for the camera, the values in EasyBuilder will update immediately.



is8402_FrostPDX - Network Settings

Host Name: is8402_FrostPDX

☑ Use DHCP Server

IP Address: 10 . 25 . 80 . 44

Subnet Mask: 255 . 255 . 254 . 0

Default Gateway:

DNS Server:

Domain Name:

DHCP Timeout:

Telnet

Telnet Port 23

Link

Link Speed Auto-Negotiate

Current Speed 100 Mbps, Full Duple

Protocol Services

iQ Sensor Solution

☐ Enable

Industrial Ethernet Protocols

● EtherNet/IP

○ PROFINET

Ethernet/IP Settings

☑ Enable User Data Bypass

b) If the Enable User Data Bypass is **NOT** enabled (unchecked which is the default state) in the Network Setting Dialog for the camera, then:

i) Set the User Data tag, SetUserData bit is a named tag, for this example it would be InSight_Top:O.Control.SetUserData.

ii) Clear the User Data tag above once you receive the User Data Ack bit. SetUserDataAck bit is a named tag, for this example it would be InSIght_Top:I.Status. SetUserDataAck.

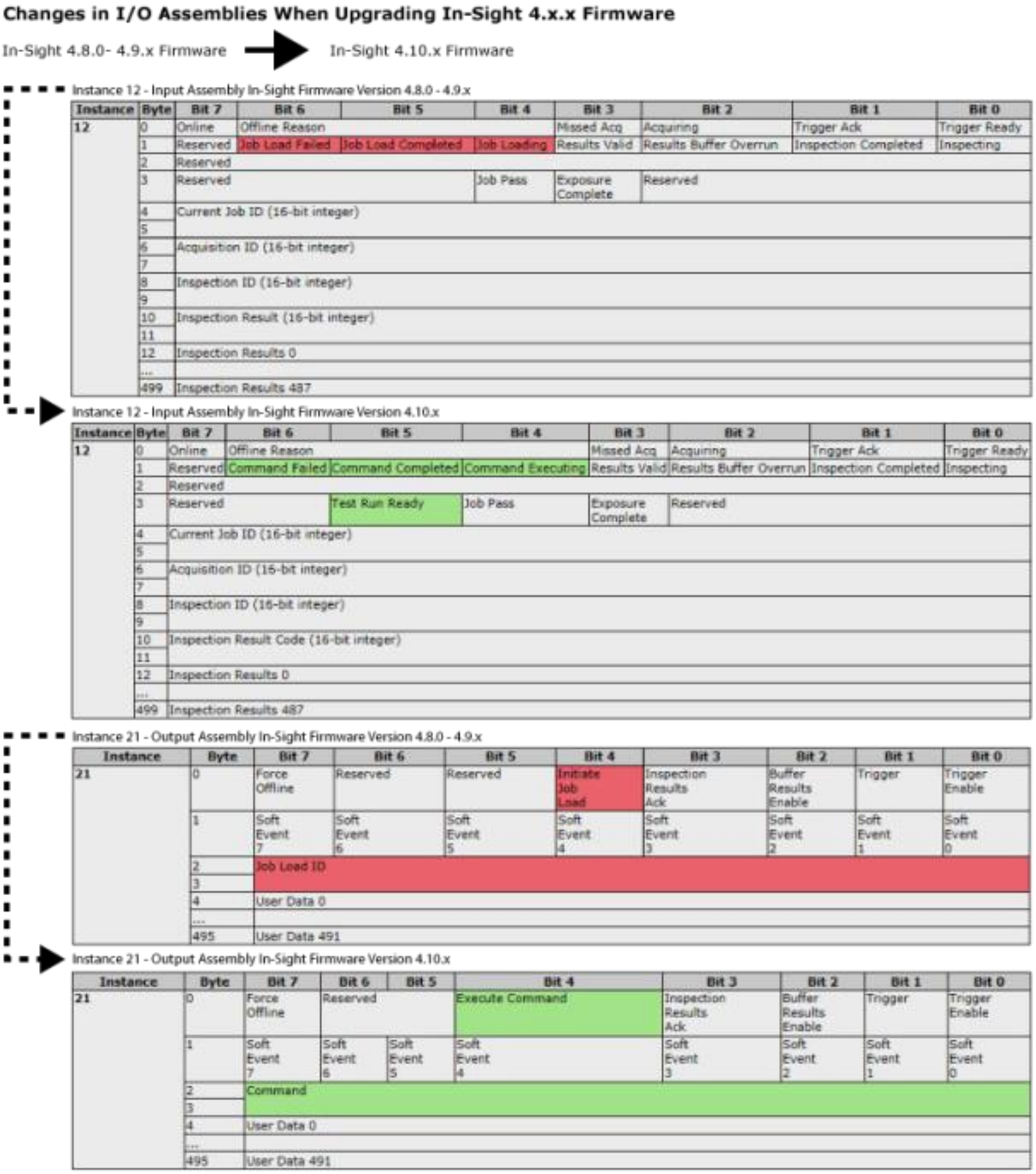
Input/Output Assembly Changes

EtherNet/IP Communication

This topic provides a graphical representation of changes to the Input and Output Assembly Objects when updating In-Sight firmware or moving between In-Sight vision system platforms.

- "Input/Output Assembly Changes When Upgrading In-Sight 4.x.x Firmware"
- Input/Output Assembly Changes When Migrating to In-Sight Vision Systems Running In-Sight 5.x.x Firmware

Input/Output Assembly Changes When Upgrading In-Sight 4.x.x Firmware



Changes in I/O Assemblies When Upgrading In-Sight 4.x.x Firmware

In-Sight 4.8.0- 4.9.x Firmware → In-Sight 4.10.x Firmware

Instance 12 - Input Assembly In-Sight Firmware Version 4.8.0 - 4.9.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12	0	Online	Offline Reason			Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	🟥 Job Load Failed	🟥 Job Load Completed	🟥 Job Loading	Results Valid	Results Buffer Overrun	Inspection Completed	Inspecting
	2	Reserved							
	3	Reserved			Job Pass	Exposure Complete	Reserved		
	4	Current Job ID (16-bit integer)							
	5								
	6	Acquisition ID (16-bit integer)							
	7								
	8	Inspection ID (16-bit integer)							
	9								
	10	Inspection Result (16-bit integer)							
	11								
	12	Inspection Results 0							
	...								
	499	Inspection Results 487							

Instance 12 - Input Assembly In-Sight Firmware Version 4.10.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12	0	Online	Offline Reason			Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	🟩 Command Failed	🟩 Command Completed	🟩 Command Executing	Results Valid	Results Buffer Overrun	Inspection Completed	Inspecting
	2	Reserved							
	3	Reserved		🟩 Test Run Ready	Job Pass	Exposure Complete	Reserved		
	4	Current Job ID (16-bit integer)							
	5								
	6	Acquisition ID (16-bit integer)							
	7								
	8	Inspection ID (16-bit integer)							
	9								
	10	Inspection Result Code (16-bit integer)							
	11								
	12	Inspection Results 0							
	...								
	499	Inspection Results 487							

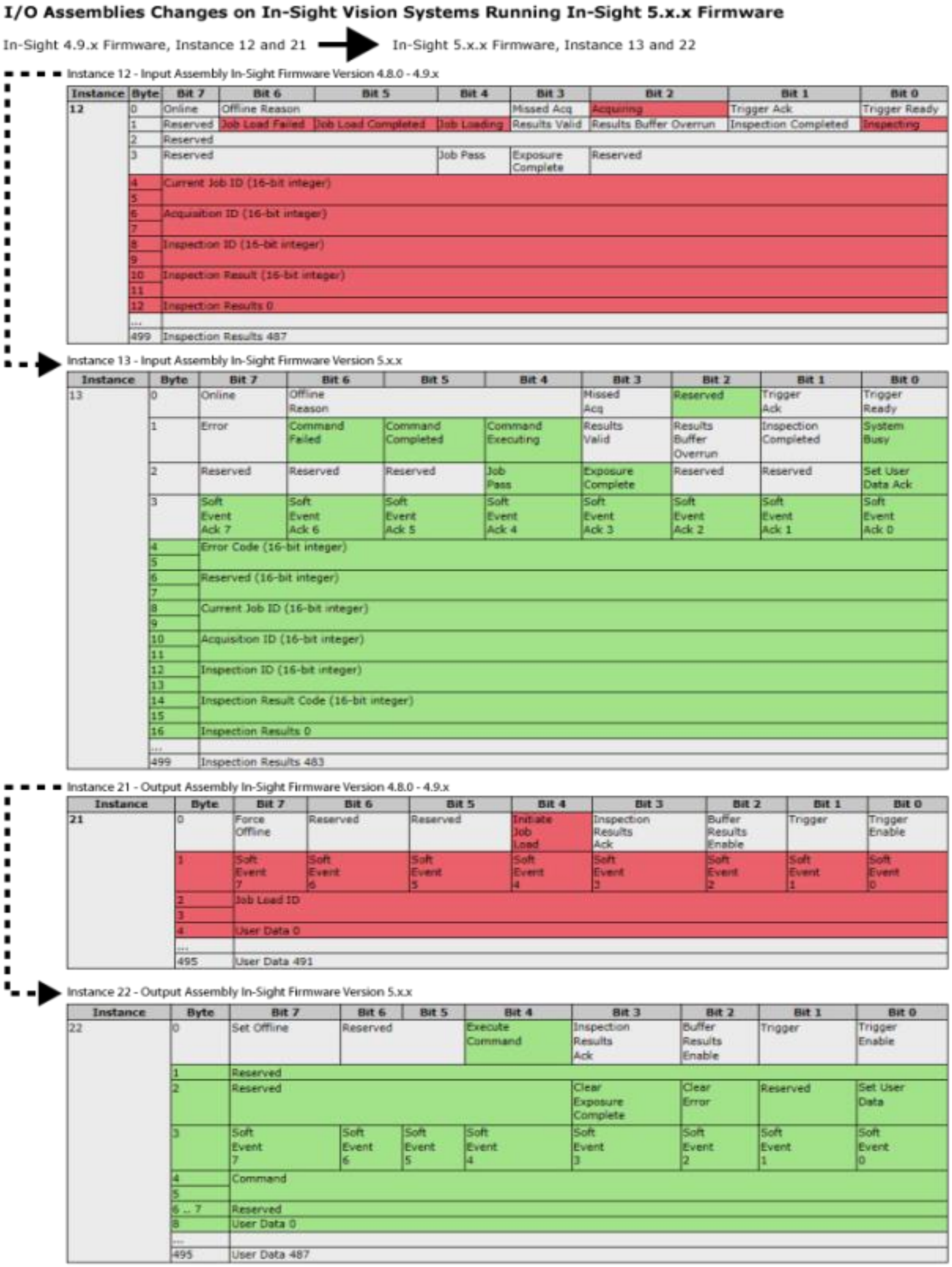
Instance 21 - Output Assembly In-Sight Firmware Version 4.8.0 - 4.9.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved	Reserved	🟥 Initiate Job Load	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	🟥 Job Load ID							
	3								
	4	User Data 0							
	...								
	495	User Data 491							

Instance 21 - Output Assembly In-Sight Firmware Version 4.10.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved		🟩 Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	🟩 Command							
	3								
	4	User Data 0							
	...								
	495	User Data 491							

Input/Output Assembly Changes When Migrating to In-Sight Vision Systems Running In-Sight 5.x.x Firmware



I/O Assemblies Changes on In-Sight Vision Systems Running In-Sight 5.x.x Firmware

In-Sight 4.9.x Firmware, Instance 12 and 21 → In-Sight 5.x.x Firmware, Instance 13 and 22

Instance 12 - Input Assembly In-Sight Firmware Version 4.8.0 - 4.9.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12	0	Online	Offline Reason			Missed Acq	🟥 Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	🟥 Job Load Failed	🟥 Job Load Completed	🟥 Job Loading	Results Valid	Results Buffer Overrun	Inspection Completed	🟥 Inspecting
	2	Reserved							
	3	Reserved			Job Pass	Exposure Complete	Reserved		
	🟥 4	🟥 Current Job ID (16-bit integer)							
	🟥 5								
	🟥 6	🟥 Acquisition ID (16-bit integer)							
	🟥 7								
	🟥 8	🟥 Inspection ID (16-bit integer)							
	🟥 9								
	🟥 10	🟥 Inspection Result (16-bit integer)							
	🟥 11								
	🟥 12	🟥 Inspection Results 0							
	...								
	499	Inspection Results 487							

Instance 13 - Input Assembly In-Sight Firmware Version 5.x.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
13	0	Online	Offline Reason			Missed Acq	🟩 Reserved	Trigger Ack	Trigger Ready
	1	Error	🟩 Command Failed	🟩 Command Completed	🟩 Command Executing	Results Valid	Results Buffer Overrun	Inspection Completed	🟩 System Busy
	2	Reserved	Reserved	Reserved	🟩 Job Pass	🟩 Exposure Complete	Reserved	Reserved	🟩 Set User Data Ack
	3	🟩 Soft Event Ack 7	🟩 Soft Event Ack 6	🟩 Soft Event Ack 5	🟩 Soft Event Ack 4	🟩 Soft Event Ack 3	🟩 Soft Event Ack 2	🟩 Soft Event Ack 1	🟩 Soft Event Ack 0
	🟩 4	🟩 Error Code (16-bit integer)							
	🟩 5								
	🟩 6	🟩 Reserved (16-bit integer)							
	🟩 7								
	🟩 8	🟩 Current Job ID (16-bit integer)							
	🟩 9								
	🟩 10	🟩 Acquisition ID (16-bit integer)							
	🟩 11								
	🟩 12	🟩 Inspection ID (16-bit integer)							
	🟩 13								
	🟩 14	🟩 Inspection Result Code (16-bit integer)							
	🟩 15								
	🟩 16	🟩 Inspection Results 0							
	...								
	499	Inspection Results 483							

Instance 21 - Output Assembly In-Sight Firmware Version 4.8.0 - 4.9.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved	Reserved	🟥 Initiate Job Load	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	🟥 1	🟥 Soft Event 7	🟥 Soft Event 6	🟥 Soft Event 5	🟥 Soft Event 4	🟥 Soft Event 3	🟥 Soft Event 2	🟥 Soft Event 1	🟥 Soft Event 0
	🟥 2	🟥 Job Load ID							
	🟥 3								
	🟥 4	🟥 User Data 0							
	...								
	495	User Data 491							

Instance 22 - Output Assembly In-Sight Firmware Version 5.x.x

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
22	0	Set Offline	Reserved		🟩 Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	🟩 1	🟩 Reserved							
	🟩 2	🟩 Reserved				🟩 Clear Exposure Complete	🟩 Clear Error	🟩 Reserved	🟩 Set User Data
	🟩 3	🟩 Soft Event 7	🟩 Soft Event 6	🟩 Soft Event 5	🟩 Soft Event 4	🟩 Soft Event 3	🟩 Soft Event 2	🟩 Soft Event 1	🟩 Soft Event 0
	🟩 4	🟩 Command							
	🟩 5								
	🟩 6 .. 7	🟩 Reserved							
	🟩 8	🟩 User Data 0							
	...								
	495	User Data 487							

EtherNet/IP Object Model and Input/Output Assembly Objects - In-Sight 5.x.x Firmware

EtherNet/IP Communications - In-Sight 5.x.x Firmware

In-Sight Communications Reference

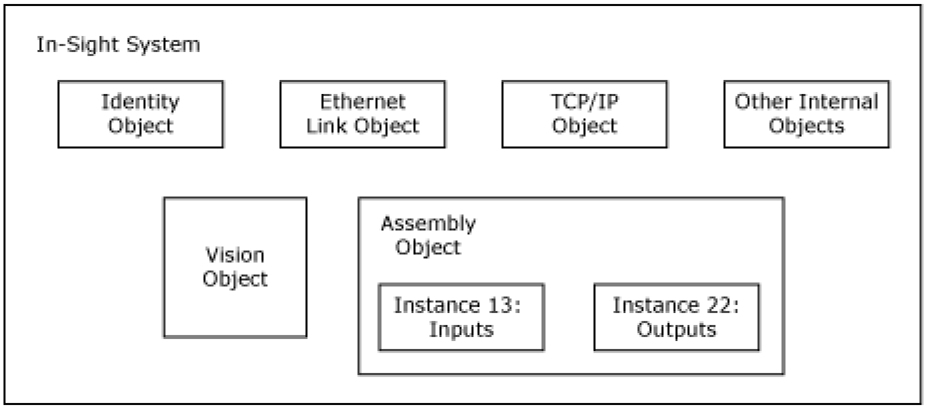
This topic covers the In-Sight Object Model for In-Sight vision systems running In-Sight firmware 5.1.0 and later.

- In-Sight Object Model
- Input/Output Assembly Objects
- Vision Object Attributes
- Vision Object Services
- EtherNet/IP Functions
- EtherNet/IP Communications

In-Sight Object Model

In EtherNet/IP networks, In-Sight vision systems act as servers, with support for both explicit and implicit I/O messaging. Data from inspections, for instance, can be transferred to clients via explicit messages or through implicit connections. Implicit connections can also be used to control acquisitions, triggers and spreadsheet events.

The Identity Object, Ethernet Link Object, TCP/IP Object and the Other Internal Objects are required by the EtherNet/IP specification. The different instances of the Assembly Object are used to exchange application data with EtherNet/IP clients. The Vision Object is defined by the In-Sight System to provide information specific to In-Sight vision systems. The details for how this occurs are provided in the Vision Object Attributes and Services section. The current object model provided by In-Sight vision systems is illustrated in the following figure:



- **Assembly Object:** The Assembly Object binds attributes of multiple objects, which allows data to and from each object to be sent or received over a single connection. Assembly Objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view. An "input" will produce data on the network and an "output" will consume data from the network.
- **Identity Object:** This object provides identification of, and general information about, the device.
- **Ethernet Link Object:** The Ethernet Link Object maintains link-specific counters and status information for an Ethernet 802.3 communications interface.
- **TCP/IP Object:** The TCP/IP Object provides a mechanism to query and possibly configure a device's TCP/IP network interface configuration. Examples of items of interest include a device's IP Address, Network Mask and Gateway Address.

- **Vision Object:** The Vision Object contains all the services and attributes specific to In-Sight vision systems. This includes acquisition, inspection, job change-over and communications with the In-Sight Explorer spreadsheet.

In-Sight Object Model

The I/O Assembly data attribute for the input and output data has the following formats:

Firmware Version	Input Assembly	Output Assembly
	Instance	Instance
5.1.0 and higher	13	22

Notes:

- If using an EDS generated profile, although the maximum size of the Input Assembly is 500 bytes, the EDS generated profile will only allow a connection of up to 496 bytes.
- When using Rockwell RSLogix Studio 5000, version 21 through 25, the In-Sight EDS generated profile’s default Input and Output Assembly sizes (496 bytes for each) cannot be edited and will default to the largest Input and Output Assembly sizes.
- The Acquisition ID will increment at the beginning of acquisition, when the Trigger Ack is generated and sent to the PLC. The Inspection ID increments when the Inspection Completed bit toggles (at which time the Results Valid is set high and the Inspecting bit goes low).
- See the Input/Output Assembly Changes topic for differences in the Input and Output Assembly tables when migrating from In-Sight vision systems running In-Sight 4.x.x firmware to vision systems running 5.x.x firmware.

I/O Assembly Data Attribute Format - Input Assemblies - Instance 13

Instance 13 is only supported on In-Sight vision systems running In-Sight firmware 5.1.0 and later.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
13	0	Online	Offline Reason			Missed Acq	Reserved	Trigger Ack	Trigger Ready
	1	Error	Command Failed	Command Completed	Command Executing	Results Valid	Results Buffer Overrun	Inspection Completed	System Busy
	2	Reserved	Reserved	Reserved	Job Pass	Exposure Complete	Reserved	Reserved	Set User Data Ack
	3	Soft Event Ack 7	Soft Event Ack 6	Soft Event Ack 5	Soft Event Ack 4	Soft Event Ack 3	Soft Event Ack 2	Soft Event Ack 1	Soft Event Ack 0
	4	Error Code (16-bit integer)							
	5								
	6	Reserved (16-bit integer)							
	7								
	8	Current Job ID (16-bit integer)							
	9								
	10	Acquisition ID (16-bit integer)							
	11								
	12	Inspection ID (16-bit integer)							
	13								

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	14	Inspection Result Code (16-bit integer)							
	15								
	16	Inspection Results 0							
	...								
	499	Inspection Results 483							

Byte 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>Online</td><td>This bit is set when the In-Sight vision system is Online, and cleared when the vision system is Offline. When the vision system is Offline, examine the *Offline Reason* field to determine the reason.</td></tr>
<tr><td>6-4</td><td>Offline Reason</td><td>This field is a 3-bit field used to identify the cause of why an In-Sight vision system is Offline:
<table>
<tr><th>Offline Reason</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Online</td><td>The vision system is Online.</td></tr>
<tr><td>1</td><td>Programming</td><td>The vision system's job is being modified.</td></tr>
<tr><td>2</td><td>Discrete Offline</td><td>A discrete signal is holding the vision system Offline.</td></tr>
<tr><td>3</td><td>Comm. Offline</td><td>A communications protocol is holding the vision system Offline.</td></tr>
</table>
Note: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.</td></tr>
<tr><td>3</td><td>Missed Acq</td><td>Set when an In-Sight vision system misses an acquisition trigger, regardless how the acquisition was triggered; cleared when an acquisition is successfully triggered.</td></tr>
<tr><td>2</td><td>Reserved</td><td>Unused.</td></tr>
<tr><td>1</td><td>Trigger Ack</td><td>Indicates when an In-Sight vision system has been triggered by the *Trigger* bit being set; this bit will stay set until the *Trigger* bit is cleared. In addition, the *Acquisition ID* can be latched to the rising edge of this bit.</td></tr>
<tr><td>0</td><td>Trigger Ready</td><td>Indicates when an In-Sight vision system can accept a new trigger via the *Trigger* bit. This field is true when the vision system is Online, the *Trigger Enable* bit is set, the AcquireImage function's Trigger parameter is set to *External* or *Industrial Ethernet*, and the vision system is not currently acquiring an image.</td></tr>
</table>

Byte 1

Bit	Name	Description
7	Error	This bit is set when an error has occurred, which is defined in the Error Code field.
6	Command Failed	This bit is set to 1 to indicate that Job Load has failed to run to completion. It is cleared when a new job is loaded by the PLC/HMI. If the job is changed through In-Sight Explorer, this bit does not change. This bit is always set prior to setting the *Command Completed* bit.
5	Command Completed	This bit is set to indicate that Job Load has completed. When a Command completes the *Command Executing* bit goes low and if the *Execute Command* bit is still high, the *Command Completed* bit is set. If the Command did not successfully complete, the *Command Failed* bit is also set. NOTE: When using Command Completed to indicate the Job Load – the command completed bit will time out after 2 seconds regardless if the job has finished loading. For jobs that take longer than 2 seconds to load – monitor the system busy bit during the job load logic to ensure the Job has finished loading before continuing.
4	Command Executing	This bit is set to 1 when Job Load is started. The *Command Completed* and *Command Failed* bits will be set prior to the falling edge of this bit.
3	Results Valid	Set when the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are set. The bit is cleared when the *Inspection Results Ack* bit is set.
2	Results Buffer	This field is set when the Buffer Results Enable bit is set and the In-Sight vision system has discarded a set of inspection results because the PLC has not acknowledged the results by setting the Inspection Results Ack bit. Up

Bit	Name	Description
	Overrun	to eight inspections are held in the vision system's buffer; therefore, this bit is set when the ninth inspection is added to the buffer. The ninth inspection, and all subsequent inspections, will be dropped until there is room in the buffer (when the results have been acknowledged out). The bit is not cleared until a valid inspection occurs and a previous inspection is not overwritten.
1	Inspection Completed	This bit is toggled upon the completion of an inspection. It is guaranteed to be toggled after the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are sent to the PLC. **Note**: Use this bit to indicate that results are ready to read. Use ResultsValid to determine when to read data.
0	System Busy	Set when the vision system is running a job, loading a job or responding to user input.

Byte 2

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7-5</td><td>Reserved</td><td>Unused.</td></tr>
<tr><td>4</td><td>Job Pass</td><td>This bit is set if the most recent job passed as configured in the Job Pass/Fail Cell Setup dialog. This bit is cleared if the job did not pass.
<b>Note</b>: The behavior of the Job Pass bit will depend on whether results buffering is disabled or enabled.
<table>
<tr><th>Buffering Is Disabled</th><th>Buffering Is Enabled</th></tr>
<tr><td>Set when an inspection is completed and the Job Pass/Fail cell indicates pass; otherwise it remains cleared. <b>Note</b>: The Job Pass bit will be valid prior to toggling the Inspection Completed bit.</td><td>Set when new results are placed in the InspectionResults attribute and the Job Pass/Fail cell indicated pass for that result set; otherwise it remains cleared. The state will not change until the results are acknowledged by setting the InspectionResultsAck attribute to True, at which time the bit will be cleared. <b>Note</b>: The Job Pass bit will be valid prior to the Results Valid bit being set.</td></tr>
</table>
</td></tr>
<tr><td>3</td><td>Exposure Complete</td><td>This bit is set upon the completion of the In-Sight vision system's exposure period, and is reset by the Clear Exposure Complete bit. This bit will be held in a reset state if the Clear Exposure Complete signal is set to High.</td></tr>
<tr><td>2-1</td><td>Reserved</td><td>Unused.</td></tr>
<tr><td>0</td><td>Set User Data Ack</td><td>This bit is set to acknowledge completion of the <i>Set User Data</i> command.</td></tr>
</table>

Byte 3

Bit	Name	Description
7	Soft Event Ack 7	These bits are used to indicate that the Soft Event command was received.
6	Soft Event Ack 6	
5	Soft Event Ack 5	
4	Soft Event Ack 4	
3	Soft Event Ack 3	
2	Soft Event Ack 2	
1	Soft Event Ack 1	

Bit	Name	Description
0	Soft Event Ack 0	

Byte 4-5

Name	Description
Error Code (16-bit Integer)	A 16-bit numeric representation of the error that has occurred:

Code	Error	Description
0x0000	No Error	No error occurred.
0x0100	Trigger set while disabled	Occurs when the Trigger bit is set while the Trigger Enable bit is cleared.
0x0101	Trigger set while Offline	Occurs when the Trigger bit is set while the vision system is Offline.
0x0400	Already Executing Error	Occurs when the Execute Command bit is set and the Executing bit is still High.
0x0401	Job load requested while Online	Occurs when a Job Load command is issued while the vision system is Online.

Byte 6-7

Name	Description
Reserved (16-bit Integer)	Unused.

Byte 8-9

Name	Description
Current Job ID (16-bit Integer)	A 16-bit integer that denotes the ID number of the currently running job on the vision system, or 65535 if the current job does not have an ID number. This field is updated when the job is changed on the vision system, regardless of method of job change.

Byte 10-11

Name	Description
Acquisition ID (16-bit Integer)	This ID increments at the beginning of an acquisition and when the *Trigger Ack* bit is set; can be used to synchronize an acquisition with its Inspection Results.

Byte 12-13

Name	Description
Inspection ID (16-bit Integer)	The acquisition ID associated with this set of results.

Byte 14-15

Name	Description
Inspection Result Code (16-bit Integer)	The inspection result code is defined by the Result Code parameter of the WriteResultsBuffer function.

Byte 16-499

Name	Description
Inspection Results (0 - 483)	This is the data that is written from In-Sight Explorer, via the WriteResultsBuffer function in the Spreadsheet View, or the Format Output Data tab in the Communications Step of the EasyBuilder GUI. The data sent will be written exactly as it appears in the In-Sight Explorer GUI, i.e. the bits appear in the same order

Name	Description
	as they are defined in the FormatOutputBuffer function (Spreadsheet) or the Format Output Data tab (EasyBuilder). **Notes**: • When the Inspection Results data array is received, RSLogix 5000 will assign a single data type to the entire array, regardless if the data that was sent contained multiple data types (i.e. the formatted data contained integers, floating point and/or bits). Therefore, if the data being sent contains different data types, copy the result (starting with the byte offset in the Inspection Results array, and the length, in bytes), into a User-Defined Data Type in RSLogix, based on the expected data type being sent. For more information, see the EXAMPLE: Getting floating point value data from an In-Sight vision system. • If the *Buffer Results Enable* bit is set, then the inspection results will remain unchanged until acknowledged by pulsing the *Inspection Results Ack* bit.

I/O Assembly Data Attribute Format - Output Assemblies - Instance 22

Instance 22 is only supported on In-Sight vision systems running In-Sight firmware 5.1.0 and later.

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
22	0	Set Offline	Reserved		Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Reserved							
	2	Reserved				Clear Exposure Complete	Clear Error	Reserved	Set User Data
	3	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	4	Command							
	5								
	6 .. 7	Reserved							
	8	User Data 0							
	...								
	495	User Data 487							

Note: In job deployment environments in which In-Sight Explorer or the VisionView application are monitoring inspections, if the job depends upon a Soft Event (configured as a Timer function, for example) to trigger a spreadsheet event, it may cause the inspection of an image to be delayed if it is triggered shortly before the acquisition cycle completes. If the job file is large (i.e. it contains many Vision Tools, such as Pattern Match, Flaw Detection and/or InspectEdge tools, in addition to additional job logic), the update required by In-Sight Explorer or VisionView may prevent an image from being inspected until the display update is queued. For applications which require exact timing (e.g. measured in the 10s of milliseconds) , this update may result in a delay in determining a pass/fail results, and thus the transmission of that result to the next station (e.g. a PLC or motion controller) in the inspection process. Therefore, in these types of application environments, Cognex recommends that Soft Events not be utilized to avoid the possibility of delayed inspections.

Byte 0

Bit	Name	Description
7	Set Offline	When this bit is set, the In-Sight vision system is taken Offline until the bit is cleared again.
6-5	Reserved	Unused.
4	Execute Command	When set, the vision system loads the job ID specified in the Command field. The signal must be held high until the *Command Completed* signal is toggled. The falling edge of this signal (prior to Command Complete) is interpreted as an abort request.

Bit	Name	Description
3	Inspection Results Ack	When the *Buffer Results Enable* bit is set, the *Inspection Results Ack* bit acknowledges that the PLC has received the *Inspection ID*, *Inspection Result* and *Inspection Results* data. The next set of inspection results is then sent to the PLC. Clearing the *Inspection Results Ack* bit causes the vision system to set the *Results Valid* bit if the buffer is not empty. If results buffering is disabled, the *Inspection Results Ack* bit must be set to clear the *Results Valid* bit.
2	Buffer Results Enable	When this bit is set, the *Inspection ID*, *Inspection Result* and *Inspection Results* fields are held constant until the *Inspection Results Ack* field has acknowledged them and been set. Up to eight inspections are held in the vision system's buffer. The In-Sight vision system will respond to the acknowledgment by clearing the *Results Valid* bit. Once the *Inspection Results Ack* field is cleared and there is a new set of rules sent to the PLC, the *Results Valid* bit will no longer be cleared. If the *Inspection Results Ack* bit is cleared and there are no more results in the vision system's buffer that are to be sent to the PLC, the *Results Valid* bit remains cleared.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: ▪ The In-Sight vision system is Online. ▪ The *Trigger Enable* bit is set. ▪ The AcquireImage function's Trigger parameter is set to *External* or *Industrial Ethernet*.
0	Trigger Enable	This field is set to enable triggering via the *Trigger* bit. Clear this bit to disable the network triggering mechanism.

Byte 1

Bit	Name	Description
7-0	Reserved	Unused.

Byte 2

Bit	Name	Description
7-4	Reserved	Unused.
3	Clear Exposure Complete	While this signal is High, the Exposure Complete status will remain reset. Once this signal is set to Low, the Exposure Complete status will be set to High on the next exposure completion.
2	Clear Error	When this bit is set, it will clear the Error and Error Code signals; the Clear Error bit should be held high until the Error bit has been cleared. If an error has been queued, clearing this bit will cause the Error and Error Code signals to be set to the next queued error code.
1	Reserved	Unused.
0	Set User Data	This command is used by the PLC to indicate to the In-Sight vision system that it should transfer the *User Data* field into a holding buffer for consumption by the vision system.

Byte 3

Bit	Name	Description
7	Soft Event 7	Allows Spreadsheet soft events to be triggered. Setting any of these bits causes the associated soft event in the Spreadsheet to be triggered.
6	Soft Event 6	
5	Soft Event 5	
4	Soft Event 4	
3	Soft Event 3	
2	Soft Event 2	
1	Soft Event 1	

Bit	Name	Description
0	Soft Event 0	

Byte 4-5

Byte	Name	Description
4 - 5	Command	This is a 16-bit integer used to indicate the Job ID number (1-999) of the job to load when the *Execute Command* bit is set by the PLC. The Command field must be held constant between the rising edge of the *Execute Command* signal and the rising edge of the Command Completed signal, or the results will be indeterminate.

Byte 6-7

Byte	Name	Description
6-7	Reserved	Unused.

Byte 8-495

Byte	Name	Description
8-495	User Data (0 - 487)	Data buffer which can be read into the spreadsheet using the ReadUserDataBuffer or ReadLatchedUserDataBuffer function. The buffer is written exactly as it appears in the PLC, with the bits appearing in the same order as they are defined in RSLogix 5000.

Mapping of I/O Assembly Data Attribute Components

The following table indicates the I/O Assembly Data attribute mapping for In-Sight vision systems:

Data Component Name	Class		Instance Number	Attribute	
	Name	Number		Name	Number
Set Offline	Vision	78 Hex	1	ForceOffline	8
Trigger Enable	Vision	78 Hex	1	AcqTriggerEnable	9
Trigger	Vision	78 Hex	1	AcqTrigger	10
BufferResultsEnable	Vision	78 Hex	1	BufferResultsEnable	13
Soft Event N	Vision	78 Hex	1	SoftTrigger bit N	18
Online	Vision	78 Hex	1	Online	6
OfflineReason	Vision	78 Hex	1	OfflineReason	7
Trigger Ready Trigger Ack Exposure Complete Missed Acquisition	Vision	78 Hex	1	AcqStatusRegister	11
User Data	Vision	78 Hex	1	User Data	12
System Busy Inspection Completed Results Buffer Overrun Results Valid	Vision	78 Hex	1	InspectionStatusRegister	14
Inspection ID Inspection Data	Vision	78 Hex	1	InspectionResults	16

Vision Object Attributes

This section describes the various attributes and services that exist in the EtherNet/IP Vision Object.

Class Code: 78 Hex -This object models all operations available to the In-Sight vision system running In-Sight firmware 5.1.0 and later.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description of Attribute
6	Get	Online	BOOL	0 = Offline 1 = Online
7	Get	OfflineReason	BYTE	0 = Online 1 = Programming 2 = Discrete Offline 3 = Comm. Offline 4-255 = Reserved
8	Set	ForceOffline	BOOL	1 = Force the In-Sight vision system Offline.
9	Set	AcqTriggerEnable	BOOL	0 = Acquisition Trigger is disabled. Trigger Ack is reset to 0. 1 = Acquisition Trigger is enabled.
10	Set	AcqTrigger	BOOL	When AcqTriggerEnable is True, the In-Sight vision system will acquire an image when AcqTrigger changes from 0 to 1.
11	Get	AcqStatusRegister	BYTE	Bit 0: Trigger Ready Bit 1: Trigger Ack Bit 2: Exposure Complete Bit 3: Missed Acquisition Bit 4-7: Reserved
12	Set	UserData	ARRAY of BYTE	User defined data that may be used as an input to the inspection.
13	Set	BufferResultsEnable	BOOL	When BufferResultsEnable is True, it enables buffering of inspection results until the ResultsAck attribute is set to True.
14	Get	InspectionStatusRegister	BYTE	Bit 0: System Busy Bit 1: Inspection Completed Bit 2: Results Buffer Overrun Bit 3: Results Valid Bit 4: Job Loading Bit 5: Job Load Completed Bit 6: Job Load Failed Bit 7: Job Pass
15	Set	InspectionResultsAck	BOOL	Acknowledges that the client received the inspection results.
16	Get	InspectionResults	STRUCT of	The last inspection results.
		InspectionID	UINT	Inspection counter.
		Reserved	UINT	Reserved field.
		InspectionResults	ARRAY of BYTE	Results from the last inspection.

Attribute ID	Access Rule	Name	Data Type	Description of Attribute
17	Get	InspectionResultString	STRING	User-defined result string.
18	Set	SoftTrigger	BYTE	Bit 0: Soft Trigger 0 Bit 1: Soft Trigger 1 Bit 2: Soft Trigger 2 Bit 3: Soft Trigger 3 Bit 4: Soft Trigger 4 Bit 5: Soft Trigger 5 Bit 6: Soft Trigger 6 Bit 7: Soft Trigger 7
19	Set	JobID	INT	The ID number of the currently loaded job, or -1 if the current job does not have an ID. Setting this value (between 1 and 154) will load the job with that associated ID.
20	Set	JobName	STRING	The file name of the currently loaded job, or an empty string if the file name is unknown. Setting this value will load the job with that associated file name.
21	Set	ClearExposureComplete	BOOL	Clears the Exposure Complete bit. The rising edge of the bit clears the Exposure Complete bit. If another Exposure Complete signal is sent while the Clear Exposure Complete bit is set, the Exposure Complete bit will be set when the Clear Exposure Complete bit is cleared.
22	Set	SoftEventAck 0-7	BYTE	Bit 0 = Soft Trigger Ack 0 Bit 1 = Soft Trigger Ack 1 Bit 2 = Soft Trigger Ack 2 Bit 3 = Soft Trigger Ack 3 Bit 4 = Soft Trigger Ack 4 Bit 5 = Soft Trigger Ack 5 Bit 6 = Soft Trigger Ack 6 Bit 7 = Soft Trigger Ack 7

Online Attributes

The Online Attributes indicate whether an In-Sight vision system is in an Online or Offline state. When the In-Sight vision system is Offline, the *OfflineReason* attribute can be used to determine the reason as to why the In-Sight vision system is Offline.

OfflineReason Attribute Values

Offline Reason	Name	Description
0	Online	The vision system is Online.
1	Programming	The vision system's job is being modified.
2	Discrete Offline	A discrete signal is holding the vision system Offline.
3	Comm. Offline	A communications protocol is holding the vision system Offline.

> **Note**: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.

The SetOffline attribute can be used to force the In-Sight vision system into an Offline state. When the SetOffline attribute is set to True, the OfflineReason attribute will be set to Comm. Offline, if the In-Sight vision system is being held Offline for no other reason.

Acquisition Attributes

> **Note**: If the In-Sight vision system will be configured to accept an acquisition trigger from a PLC/Motion Controller via a Native Mode command, Cognex recommends that the SetEvent and Wait function be utilized, with the Event code set to 8 (**SW8**). This will ensure that vision system waits for both the acquisition and inspection to be completed before sending a "complete" response back to the PLC/Motion Controller, and that previous inspection results are not being sent to the PLC/Motion Controller. The "complete" response from the vision system can also then be used to create conditional PLC logic that sends a read request for the inspection results.

The Vision Object can be triggered to acquire images by using the Acquisition Attributes. To reset the Acquisition Attributes in the Vision Object, the AcqTriggerEnable attribute must be set to False, until the AcqStatusRegister is 0. Then, the AcqTriggerEnable attribute can be set to True to enable acquisition via the Vision Object. When the Acquisition Trigger is ready to accept triggers, the Trigger Ready bit in the AcqStatusRegister will be set.

While the AcqTriggerEnable attribute is set to True, each time the Vision Object sees the AcqTrigger attribute change from 0 to 1, it will initiate an image acquisition. When setting this via implicit messaging, to guarantee that the change is seen by the Vision Object, the attribute should be held in the changed state until that same value is seen by the Trigger Ack bit of the AcqStatusRegister.

> **Note**: This is not necessary when using explicit messaging, because the change will always be seen.

To determine when an In-Sight vision system has completed an image acquisition, the Exposure Complete and Clear Exposure Complete bits should be utilized. Acquisitions must be explicitly acknowledged by explicitly acknowledging the Exposure Complete signal. If an Exposure Complete is signaled while the Clear Exposure Complete bit is HIGH, the Exposure Complete bit will be asserted when the Clear Exposure Complete bit is cleared by the PLC.

With this manner of operation, if multiple Exposure Complete signals occur, they will have the same effect as though just one signal was sent, and no errors will be reported in this scenario. This allows a PLC to detect when an acquisition was triggered from a discrete input by detecting the Exposure Complete signal.

If other methods of acquisition triggering are employed, the Exposure Complete and Missed Acquisition bits of the AcqStatusRegister will still be active.

Inspection Results Attributes

When an image is acquired as a result of an acquisition trigger, it is placed in a queue for inspection. As the inspection is running on the image, the System Busy bit of the InspectionStatusRegister is set. Once the inspection is completed, the System Busy bit is cleared and the Inspection Completed bit is toggled.

The BufferResultsEnable attribute determines how inspection results are handled in the Vision Object. If the BufferResultsEnable attribute is set to False, then the inspection results are immediately placed into the InspectionResults attribute.

If the BufferResultsEnable attribute is set to True, then the previous inspection results remain in the InspectionResults attribute until they are acknowledged by setting the InspectionResultsAck attribute to True. Once the Results Valid bit is cleared, the InspectionResultsAck attribute should be set to False to allow new results to be placed in to the InspectionResults attribute. This can be used to establish a handshake to obtain results between the Vision Object and the client.

Behavior of InspectionStatusRegister

Inspection Status Register Bit	Bit Name	Results If Buffering Is Disabled	Results If Buffering Is Enabled
0	System Busy	Set when the vision system is running a job, loading a job or responding to user input.	Set when the vision system is running a job, loading a job or responding to user input.
1	Inspection Completed	This bit is toggled upon the completion of an inspection. It is guaranteed to be toggled after the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are sent to the PLC.	This bit is toggled upon the completion of an inspection. It is guaranteed to be toggled after the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are sent to the PLC.

Inspection Status Register Bit	Bit Name	Results If Buffering Is Disabled	Results If Buffering Is Enabled
2	Results Buffer Overrun	Always cleared.	Set when inspection results could not be queued because the client has failed to acknowledge a previous result. Cleared when the inspection result is successfully queued.
3	Results Valid	Set when the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are set. The bit is cleared when the *Inspection Results Ack* bit is set. **Note**: If job processing is enabled to occur in overlapped mode, either the *Buffer Results Enable* bit should be enabled/set, or the *Inspection Completed* bit should be used to latch the inspection results.	Set when the *Inspection Count*, *Inspection Result Code*, *Inspection Results* and/or *Job Pass* bits are set. The bit is cleared when the *Inspection Results Ack* bit is set. **Note**: If job processing is enabled to occur in overlapped mode, either the *Buffer Results Enable* bit should be enabled/set, or the *Inspection Completed* bit should be used to latch the inspection results.
7	Job Pass	Set when an inspection is completed and the Job Pass/Fail cell indicates pass; otherwise it remains cleared. **Note**: The Job Pass bit will be valid prior to toggling the Inspection Completed bit.	Set when new results are placed in the InspectionResults attribute and the Job Pass/Fail cell indicated pass for that result set; otherwise it remains cleared. The state will not change until the results are acknowledged by setting the InspectionResultsAck attribute to True, at which time the bit will be cleared. **Note**: The Job Pass bit will be valid prior to the Results Valid bit being set.

Job Attributes

The Vision Object can load jobs stored in the non-volatile flash memory of an In-Sight vision system by setting the JobID and JobName attributes. If the load fails, the set operation will also fail.

Vision Object Services

Common Services

The Vision Object provides the following Common Services:

Service Code	Service Name	Description of Service
05 Hex	Reset	Resets the Vision Object
0E Hex	Get_Attribute_Single	Returns the contents of the specified attribute.
10 Hex	Set_Attribute_Single	Modifies an attribute value.

Reset Service

The Reset Service will reset and disable the acquisition triggering system in the Vision Object. It will also disable and clear the inspection results buffering system in the Vision Object.

Object Specific Services

The Vision Object provides the following Object Specific Services:

Service Code	Service Name	Description of Service
32 Hex	Acquire	Triggers a single acquisition.
33 Hex	Reserved	Deprecated.
34 Hex	SendNativeCmd	Sends a Native Mode command to the In-Sight vision system.

Service Code	Service Name	Description of Service
35 Hex	GetInspectionResults	Gets the InspectionResults attribute.
36 Hex	Transfer User Data	Transfers data into a portion of the User Data Holding Buffer.
37 Hex	Set User Data	Indicates to the vision system that it should latch the User Data Holding Buffer into the User Data field to give the In-Sight access to the user data.

Acquire Service

The Acquire Service of the Vision Object will cause an acquisition to be triggered, if the acquisition system is ready to acquire an image. If the acquisition could not be triggered, then the Missed Acquisition bit of the AcqStatusRegister will be set until the next successful acquisition.

If the command fails, the Acquire Service will return the vendor specific error code 1F.

SendNativeCmd Service

The SendNativeCmd Service sends any documented Native Mode command string to the In-Sight vision system. The request data for the SendNativeCmd should include the Native Mode command string that is to be sent to the In-Sight vision system; the reply data will contain the string result of the Native Mode command.

SendNativeCmd Request Data Format

Name	Type	Description
Command	STRING	The Native Mode command to process.

SendNativeCmd Response Data Format

Name	Type	Description
Result	STRING	The string result of the Native Mode command.

If the command fails, the service will return one of the following vendor specific error codes:

Error Code	Description	Native Mode Return Code
1	Bad Command	-
4	No Answer - The system is too busy.	
100	Command Failed	0
101	Command Cannot Be Executed	-1
102		-2
103		-3
104		-4
105		-5
106		-6

GetInspectionResults Service

The GetInspectionResults Service (code 0x35) reads data from the InspectionResults attribute of the Vision Object. This service can be used to read data from the InspectionResults attribute from a device which does not support receiving messages of 260 bytes.

GetInspectionResults Request Data Format

Name	Type	Description
Size	UINT	The number of bytes of the InspectionResults attribute to read.
Offset	UINT	(Optional) The offset into the InspectionResults attribute. This specifies the first byte of the InspectionResults attribute to read.

GetInspectionResults Response Data Format

Name	Type	Description
Data	Array of BYTE	The InspectionResults attribute data.

Transfer User Data Command

This command is used to transfer data into a portion of the User Data Holding Buffer. The User Data Holding Buffer will be latched to the User Data field when the Set User Data command is issued.

Request Format

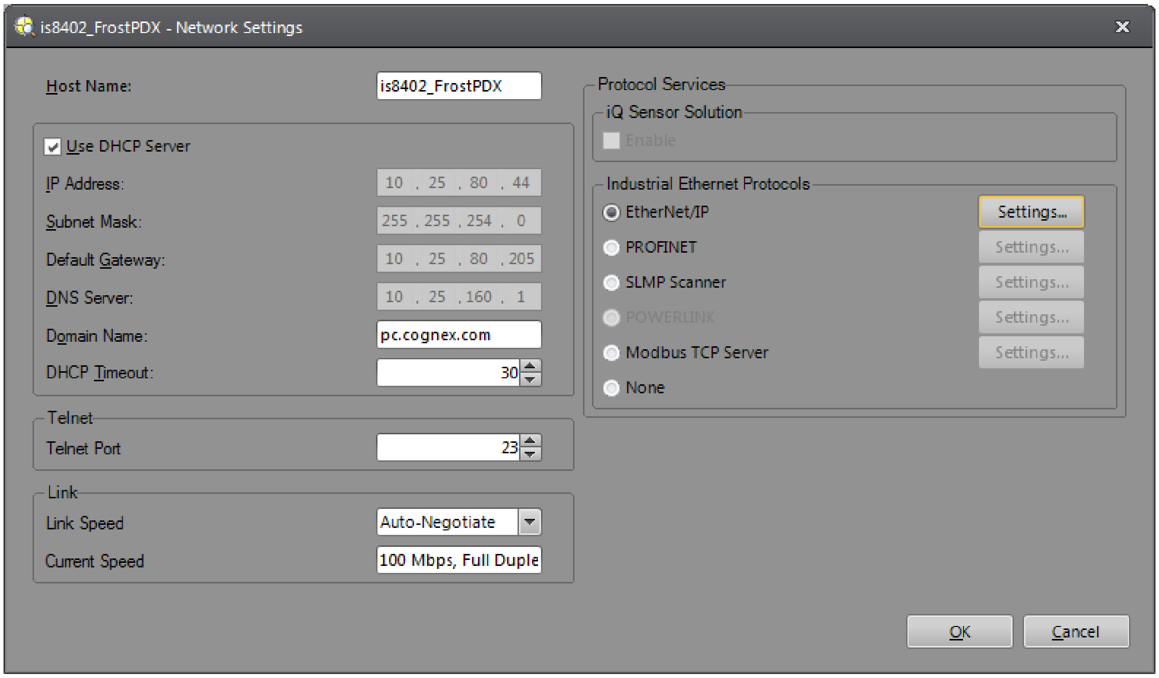
Name	Type	Description
Size	UINT	The number of bytes of the User Data field to write.
Offset	UINT	The offset into the User Data field.
Data	Array of BYTE	User Data to write into the User Data field.

Set User Data Command

This command is used by the PLC to indicate to the vision system that it should latch the User Data Holding Buffer into the User Data field to give the In-Sight access to the user data.

If this additional **User Data** handshake is not needed, and the communication should have the same behavior as the assembly IO on a 4.xx camera, this User Data Command can be bypassed by:

1) Logging onto the camera
2) From the main menu bar, select Sensor ->Network Settings
3) Select EtherNet/IP in the Industrial Ethernet Protocols section:



is8402_FrostPDX - Network Settings

Host Name: is8402_FrostPDX

☑ Use DHCP Server

IP Address: 10 . 25 . 80 . 44

Subnet Mask: 255 . 255 . 254 . 0

Default Gateway: 10 . 25 . 80 . 205

DNS Server: 10 . 25 . 160 . 1

Domain Name: pc.cognex.com

DHCP Timeout: 30

Telnet

Telnet Port 23

Link

Link Speed Auto-Negotiate

Current Speed 100 Mbps, Full Duple

Protocol Services

iQ Sensor Solution

☐ Enable

Industrial Ethernet Protocols

● EtherNet/IP

○ PROFINET

○ SLMP Scanner

○ POWERLINK

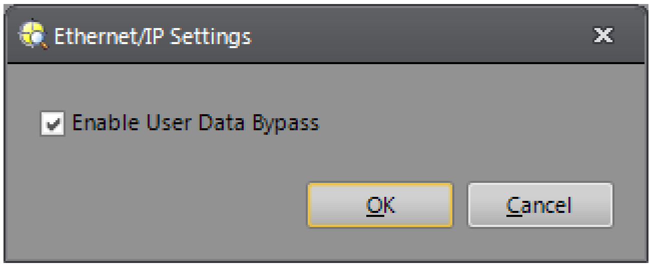
○ Modbus TCP Server

○ None

Cognex

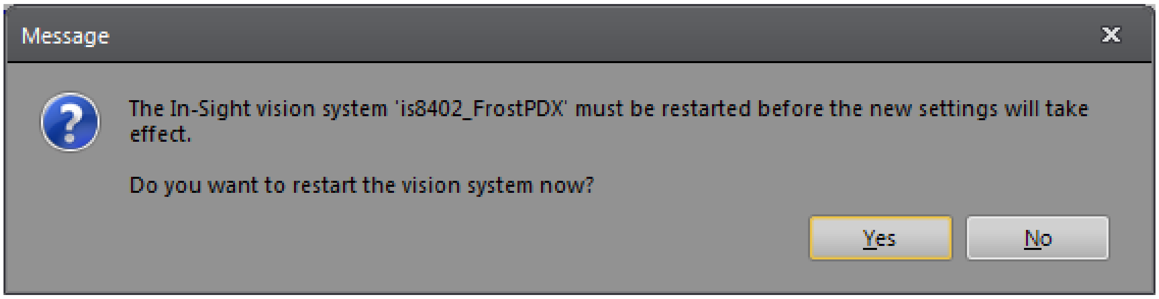
4) Select the enabled Settings button next to the EtherNet/IP selection.

5) Check ON Enable User Data Bypass:



☑ Enable User Data Bypass

6) Restart the vision system



The In-Sight vision system 'is8402_FrostPDX' must be restarted before the new settings will take effect.

Do you want to restart the vision system now?

EtherNet/IP Object Model and Input/Output Assembly Objects - In-Sight 4.x.x Firmware

EtherNet/IP Communications - In-Sight 4.x.x Firmware

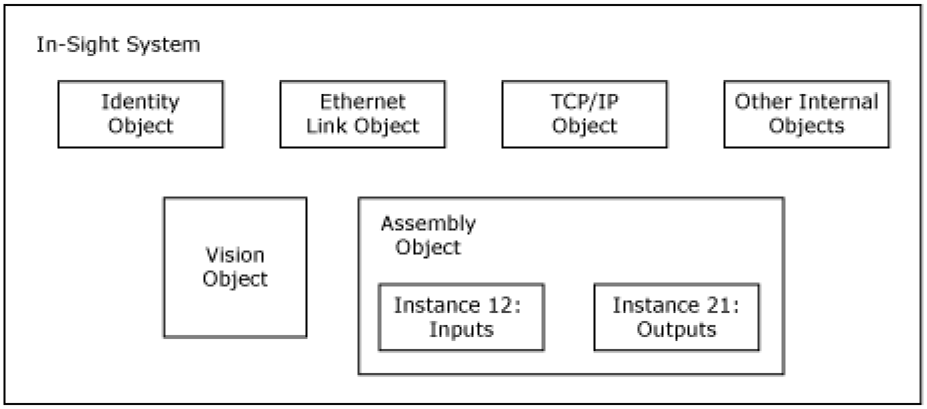
This topic covers the In-Sight Object Model for In-Sight vision systems running In-Sight 4.x.x and earlier firmware.

- In-Sight Object Model
- Input and Output Assembly Objects
- Vision Object Attributes
- Vision Object Services
- EtherNet/IP Functions
- EtherNet/IP Communications

In-Sight Object Model

In EtherNet/IP networks, In-Sight vision systems act as servers, with support for both explicit and implicit I/O messaging. Data from inspections, for instance, can be transferred to clients via explicit messages or through implicit connections. Implicit connections can also be used to control acquisitions, triggers and spreadsheet events.

The Identity Object, Ethernet Link Object, TCP/IP Object and the Other Internal Objects are required by the EtherNet/IP specification. The different instances of the Assembly Object are used to exchange application data with EtherNet/IP clients. The Vision Object is defined by the In-Sight System to provide information specific to In-Sight vision systems. The details for how this occurs are provided in the Vision Object Attributes and Services section. The current object model provided by In-Sight vision systems is illustrated in the following figure:



- **Assembly Object:** The Assembly Object binds attributes of multiple objects, which allows data to and from each object to be sent or received over a single connection. Assembly Objects can be used to bind input data or output data. The terms "input" and "output" are defined from the network's point of view. An "input" will produce data on the network and an "output" will consume data from the network.
- **Identity Object:** This object provides identification of, and general information about, the device.
- **Ethernet Link Object:** The Ethernet Link Object maintains link-specific counters and status information for an Ethernet 802.3 communications interface.
- **TCP/IP Object:** The TCP/IP Object provides a mechanism to query and possibly configure a device's TCP/IP network interface configuration. Examples of items of interest include a device's IP Address, Network Mask and Gateway Address.
- **Vision Object:** The Vision Object contains all of the services and attributes specific to In-Sight vision systems. This includes acquisition, inspection, job change-over and communications with the In-Sight Explorer spreadsheet.

Input and Output Assembly Objects

The I/O Assembly data attribute for the input and output data has the following formats:

Firmware Version	Input Assembly Instance	Output Assembly Instance
4.1.0 - 4.7.4	11	21
4.8.0 - 4.9.x	12	21
4.10.x	12	21

Notes:

- If using the Input Assembly Instance 12 that contains the Job Pass, Exposure Complete and Current Job ID (16-bit integer) blocks, the In-Sight Add-On Profile (AOP), with Major Revision 10 must be installed on the PC with RSLogix.
- If using an EDS generated profile, although the maximum size of the Input Assembly is 500 bytes, the EDS generated profile will only allow a connection of up to 496 bytes.
- When using Rockwell RSLogix Studio 5000, version 21 through 25, the In-Sight EDS generated profile's default Input and Output Assembly sizes (496 bytes for each) cannot be edited and will default to the largest Input and Output Assembly sizes.
- The Acquisition ID will increment at the end of acquisition and moves along with the image to inspection subsystem. It can be updated in the PLC depending on the RPI before the inspection is completed. The Inspection ID increments when the Inspection Completed bit toggles (at which time the Results Valid is set high and the Inspecting bit goes low).
- See the Input/Output Assembly Changes topic for differences in the Input and Output Assembly tables when upgrading In-Sight firmware from In-Sight 4.9.x to 4.10.x.

I/O Assembly Data Attribute Format - In-Sight Firmware Version 4.10.x

Input Assemblies - Instance 12

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12	0	Online	Offline Reason			Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	Command Failed	Command Completed	Command Executing	Results Valid	Results Buffer Overrun	Inspection Completed	Inspecting
	2	Reserved							
	3	Reserved		Test Run Ready	Job Pass	Exposure Complete	Reserved		
	4	Current Job ID (16-bit integer)							
	5								
	6	Acquisition ID (16-bit integer)							
	7								
	8	Inspection ID (16-bit integer)							
	9								
	10	Inspection Result Code (16-bit integer)							
	11								
	12	Inspection Results 0							
	...								
	499	Inspection Results 487							

Byte 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>Online</td><td>This bit is set when the In-Sight vision system is Online, and cleared when the vision system is Offline. When the vision system is Offline, examine the <i>Offline Reason</i> field to determine the reason.</td></tr>
<tr><td>6 - 4</td><td>Offline Reason</td><td>This field is a 3-bit field used to identify the cause of why an In-Sight vision system is Offline:
<table>
<tr><th>Offline Reason</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Online</td><td>The vision system is Online.</td></tr>
<tr><td>1</td><td>Programming</td><td>The vision system's job is being modified.</td></tr>
<tr><td>2</td><td>Discrete Offline</td><td>A discrete signal is holding the vision system Offline.</td></tr>
<tr><td>3</td><td>Comm. Offline</td><td>A communications protocol is holding the vision system Offline.</td></tr>
</table>
<b>Note</b>: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.</td></tr>
<tr><td>3</td><td>Missed Acq</td><td>Set when an In-Sight vision system misses an acquisition trigger, regardless how the acquisition was triggered; cleared when an acquisition is successfully triggered.</td></tr>
<tr><td>2</td><td>Acquiring</td><td>Set when the In-Sight vision system is currently acquiring an image; either by setting the Trigger bit or by an external hardware trigger.</td></tr>
<tr><td>1</td><td>Trigger Ack</td><td>Indicates when the In-Sight vision system has been triggered by setting the Trigger bit; this bit will stay set until the Trigger bit is cleared.</td></tr>
<tr><td>0</td><td>Trigger Ready</td><td>Indicates when the In-Sight vision system can accept a new trigger. This bit is high when the vision system is Online, the Trigger Enable bit is set and the vision system is not currently acquiring an image.</td></tr>
</table>

Byte 1

Bit	Name	Description
7	Reserved	Unused.
6	Command Failed (4.10.x)	This bit is set to 1 to indicate that the TestRun execution or Job Load has failed to run to completion. It is cleared when a new TestRun sequence is executed or a new job is loaded by the PLC/HMI. If the job is changed through In-Sight Explorer, this bit does not change.
5	Command Completed (4.10.x)	This bit is toggled to indicate that the TestRun execution or Job Load has completed. **Note**: An attempt to execute TestRun on a Job without a TestRun configuration will result in the CommandCompleted being toggled and the CommandFailed being set. TestRun should not be executed if TestRunReady is not set.
4	Command Executing (4.10.x)	This bit is set to 1 when a TestRun or Job Load is started. Cleared when TestRun execution completes. The Command Completed and Command Failed bits will be set prior to the falling edge of this bit.
3	Results Valid	Set when the Inspection ID, Inspection Result and Inspection Results fields are valid. **Note**: This bit indicates that inspection results are ready to be read.
2	Results Buffer Overrun	This field is set when the Buffer Results Enable bit is set and the In-Sight vision system has discarded a set of inspection results because the PLC has not acknowledged the results by setting the Inspection Results Ack bit.
1	Inspection Completed	This bit is toggled upon the completion of an inspection. **Note**: This bit indicates that results are ready to read when Buffer Results Enable bit is not enabled/set.
0	Inspecting	This bit is set when the In-Sight vision system is executing spreadsheet functions.

Byte 2

Name	Description
Reserved	Unused.

Byte 3

Bit	Name	Description
7 - 6	Reserved	Unused.
5	TestRun Ready (4.10.x)	This bit is set to 1 when the Vision System has a valid TestRun configuration. This signal is cleared while TestRun executes (regardless of the TestRun initiator) and returns to 1 after execution completes. **Note**: This bit is signaled regardless of whether the vision system is online or offline.
4	Job Pass	This bit is set if the most recent job passed as configured in the Job Pass/Fail Cell Setup dialog. This bit is cleared if the job did not pass. **Note**: To use this bit, use Major Revision 10 or higher.
3	Exposure Complete	This bit is set upon the completion of the In-Sight vision system's exposure period, and is cleared when an acquisition is triggered. This bit will be held in a reset state if the Clear Exposure Complete signal is set to High. **Note**: To use this bit, use Major Revision 10 or higher.
2 - 0	Reserved	Unused.

Byte 4-5

Name	Description
Current Job ID (16-bit Integer)	A 16-bit integer that denotes the ID number of the currently running job on the vision system, or 65535 if the current job does not have an ID number. This field is updated when the job is changed on the vision system, regardless of method of job change.

Byte 6-7

Name	Description
Acquisition ID (16-bit Integer)	This ID increments on the completion of every acquisition regardless of the trigger source, and can be used to synchronize an Acquisition with its Inspection Results.

Byte 8-9

Name	Description
Inspection ID (16-bit Integer)	The acquisition ID associated with this set of results. This value increments once each inspection cycle at the arrival of new results.

Byte 10-11

Name	Description
Inspection Result Code (16-bit integer) (4.10.x)	Indicates the result of the latest TestRun execution. If all tests pass, the bit 0 will be set. If one or more tests do not pass, or if there is a problem with the Setup or Cleanup during the TestRun, the bit 0 will be cleared. • This field is only valid following a successful TestRun. • The results in this field can become invalidated if a Job without a TestRun is loaded, another TestRun job is executed, a TestRun is in progress, TestRun or job load has failed or a TestRun is aborted. • If the vision system is currently set to Online, the content of this field is driven by the spreadsheet and does not have any TestRun meaning.

Byte 12-499

Name	Description

Name	Description
Inspection Results (0 - 487)	This is the data that is written from In-Sight Explorer, via the WriteEIPBuffer function in the Spreadsheet View, or the Format Output Data tab in the Communications Step of the EasyBuilder GUI. The data sent will be written exactly as it appears in the In-Sight Explorer GUI, i.e. the bits appear in the same order as they are defined in the FormatOutputBuffer function (Spreadsheet) or the Format Output Data tab (EasyBuilder). **Note**: When the Inspection Results data array is received, RSLogix 5000 will assign a single data type to the entire array, regardless if the data that was sent contained multiple data types (i.e. the formatted data contained integers, floating point and/or bits). Therefore, if the data being sent contains different data types, copy the particular result (starting with the byte offset in the Inspection Results array, and the length, in bytes), into a User-Defined Data Type in RSLogix, based on the expected data type being sent. For more information, see the EXAMPLE: Getting floating point value data from an In-Sight vision system.

Output Assemblies - Instance 21

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved		Execute Command	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	Command							
	3								
	4	User Data 0							
	...								
	495	User Data 491							

Byte 0

Bit	Name	Description
7	Force Offline	When this bit is set, the In-Sight vision system is taken Offline until the bit is cleared again.
6 - 5	Reserved	Unused.
4	Execute Command (4.10.x)	When set, the vision system loads the job ID specified in the Command field. When using TestRun, the rising edge of this signal will execute command to start TestRun. The signal must be held high until the Command Completed signal is toggled. The falling edge of this signal (prior to Command Complete) is interpreted as an abort request.
3	Inspection Results Ack	When the Buffer Results Enable bit is set, the Inspection Results Ack bit acknowledges that the PLC has received the Inspection ID, Inspection Result and Inspection Results data.
2	Buffer Results Enable	Set this bit to hold the Inspection ID, Inspection Result and Inspection Results fields constant while the ResultsValid bit is set. Set the Inspection Results Ack bit to acknowledge them. The In-Sight vision system responds to the acknowledgement by clearing the Results Valid bit. Once the Inspection Results Ack field is cleared, Results Valid remains cleared until new results are sent to the PLC.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: ▪ The In-Sight vision system is Online. ▪ The *Trigger Enable* bit is set. ▪ The AcquireImage function's Trigger parameter is set to Network, External or Industrial Ethernet.

Bit	Name	Description
0	Trigger Enable	This field is set to enable triggering via the Trigger bit. Clear this field to disable the EtherNet/IP triggering mechanism. This bit only affects triggering done via Ethernet/IP, and does not affect any other triggering mode.

Byte 1

Bit	Name	Description
7	Soft Event 7	Allows Spreadsheet soft events to be triggered. Setting any of these bits causes the associated soft event in the Spreadsheet to be triggered.
6	Soft Event 6	
5	Soft Event 5	
4	Soft Event 4	
3	Soft Event 3	
2	Soft Event 2	
1	Soft Event 1	
0	Soft Event 0	

Byte 2-3

Byte	Name	Description
2 - 3	Command (4.10.x)	This value indicates the TestRun sequence to execute or the ID number (1-999) of the job to load when the Execute Command bit is set by the PLC. 0x1007 = Execute TestRun 0x0000 – 0x03E7 = Job IDs

Byte 4-495

Byte	Name	Description
4 - 495	User Data (0 - 491)	This data is sent to the In-Sight Explorer spreadsheet via the ReadEIPBuffer function. The buffer is written exactly as it appears in the PLC, with the bits appearing in the same order as they are defined in RSLogix 5000.

I/O Assembly Data Attribute Format - In-Sight Firmware Version 4.8.0 - 4.9.x

Input Assemblies - Instance 12

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12	0	Online	Offline Reason			Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	Job Load Failed	Job Load Completed	Job Loading	Results Valid	Results Buffer Overrun	Inspection Completed	Inspecting
	2	Reserved							
	3	Reserved			Job Pass	Exposure Complete	Reserved		
	4	Current Job ID (16-bit integer)							
	5								
	6	Acquisition ID (16-bit integer)							
	7								
	8	Inspection ID (16-bit integer)							
	9								
	10	Inspection Result (16-bit integer)							
	11								
	12	Inspection Results 0							
	...								
	499	Inspection Results 487							

Byte 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>Online</td><td>This bit is set when the In-Sight vision system is Online, and cleared when the vision system is Offline. When the vision system is Offline, examine the <i>Offline Reason</i> field to determine the reason.</td></tr>
<tr><td>6 - 4</td><td>Offline Reason</td><td>This field is a 3-bit field used to identify the cause of why an In-Sight vision system is Offline:
<table>
<tr><th>Offline Reason</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Online</td><td>The vision system is Online.</td></tr>
<tr><td>1</td><td>Programming</td><td>The vision system's job is being modified.</td></tr>
<tr><td>2</td><td>Discrete Offline</td><td>A discrete signal is holding the vision system Offline.</td></tr>
<tr><td>3</td><td>Comm. Offline</td><td>A communications protocol is holding the vision system Offline.</td></tr>
</table>
<b>Note</b>: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.</td></tr>
<tr><td>3</td><td>Missed Acq</td><td>Set when an In-Sight vision system misses an acquisition trigger, regardless how the acquisition was triggered; cleared when an acquisition is successfully triggered.</td></tr>
<tr><td>2</td><td>Acquiring</td><td>Set when the In-Sight vision system is currently acquiring an image; either by setting the Trigger bit or by an external hardware trigger.</td></tr>
<tr><td>1</td><td>Trigger Ack</td><td>Indicates when the In-Sight vision system has been triggered by setting the Trigger bit; this bit will stay set until the Trigger bit is cleared.</td></tr>
<tr><td>0</td><td>Trigger Ready</td><td>Indicates when the In-Sight vision system can accept a new trigger. This bit is high when the vision system is Online, the Trigger Enable bit is set and the vision system is not currently acquiring an image.</td></tr>
</table>

Byte 1

Bit	Name	Description
7	Reserved	Unused.
6	Job Load Failed	This bit is set when the last job load attempt failed. It is cleared the next time a job is successfully loaded. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
5	Job Load Completed	This bit is toggled upon the completion of a job load operation. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
4	Job Loading	This bit is set when loading a new job. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
3	Results Valid	Set when the Inspection ID, Inspection Result and Inspection Results fields are valid. **Note**: This bit indicates that inspection results are ready to be read.
2	Results Buffer Overrun	This field is set when the Buffer Results Enable bit is set and the In-Sight vision system has discarded a set of inspection results because the PLC has not acknowledged the results by setting the Inspection Results Ack bit.
1	Inspection Completed	This bit is toggled upon the completion of an inspection. **Note**: This bit indicates that results are ready to read when Buffer Results Enable bit is not enabled/set.
0	Inspecting	This bit is set when the In-Sight vision system is executing spreadsheet functions.

Byte 2

Name	Description
Reserved	Unused.

Byte 3

Bit	Name	Description
7 - 6	Reserved	Unused.
5	Reserved	Unused.
4	Job Pass	This bit is set if the most recent job passed as configured in the Job Pass/Fail Cell Setup dialog. This bit is cleared if the job did not pass. **Note**: To use this bit, use Major Revision 10 or higher.
3	Exposure Complete	This bit is set upon the completion of the In-Sight vision system's exposure period, and is cleared when an acquisition is triggered. This bit will be held in a reset state if the Clear Exposure Complete signal is set to High. **Note**: To use this bit, use Major Revision 10 or higher.
2 - 0	Reserved	Unused.

Byte 4-5

Name	Description
Current Job ID (16-bit Integer)	A 16-bit integer that denotes the ID number of the currently running job on the vision system, or 65535 if the current job does not have an ID number. This field is updated when the job is changed on the vision system, regardless of method of job change.

Byte 6-7

Name	Description
Acquisition ID (16-bit Integer)	This ID increments on the completion of every acquisition regardless of the trigger source, and can be used to synchronize an Acquisition with its Inspection Results.

Byte 8-9

Name	Description
Inspection ID (16-bit Integer)	The acquisition ID associated with this set of results. This value increments once each inspection cycle at the arrival of new results.

Byte 10-11

Name	Description
Inspection Result (16-bit Integer)	The pass/fail code that is associated with this set of results. This value cannot be set and is always 0.

Byte 12-499

Name	Description
Inspection Results (0 - 487)	This is the data that is written from In-Sight Explorer, via the WriteEIPBuffer function in the Spreadsheet View, or the Format Output Data tab in the Communications Step of the EasyBuilder GUI. The data sent will be written exactly as it appears in the In-Sight Explorer GUI, i.e. the bits appear in the same order as they are defined in the FormatOutputBuffer function (Spreadsheet) or the Format Output Data tab (EasyBuilder). **Note**: When the Inspection Results data array is received, RSLogix 5000 will assign a single data type to the entire array, regardless if the data that was sent contained multiple data types (i.e. the formatted data contained integers, floating point and/or bits). Therefore, if the data being sent contains different data types, copy the result (starting with the byte offset in the Inspection Results array, and the length, in bytes), into a User-Defined Data Type in RSLogix, based on the expected data type being sent. For more information, see the EXAMPLE: Getting floating point value data from an In-Sight vision system.

Output Assemblies - Instance 21

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved	Reserved	Initiate Job Load	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	Job Load ID							
	3								
	4	User Data 0							
	...								
	495	User Data 491							

Byte 0

Bit	Name	Description
7	Force Offline	When this bit is set, the In-Sight vision system is taken Offline until the bit is cleared again.

Bit	Name	Description
6 - 5	Reserved	Unused.
4	Initiate Job Load (4.9.x and earlier)	When set, the vision system loads the job ID specified in the Job Load ID field.
3	Inspection Results Ack	When the Buffer Results Enable bit is set, the Inspection Results Ack bit acknowledges that the PLC has received the Inspection ID, Inspection Result and Inspection Results data.
2	Buffer Results Enable	Set this bit to hold the Inspection ID, Inspection Result and Inspection Results fields constant while the ResultsValid bit is set. Set the Inspection Results Ack bit to acknowledge them. The In-Sight vision system responds to the acknowledgement by clearing the Results Valid bit. Once the Inspection Results Ack field is cleared, Results Valid remains cleared until new results are sent to the PLC.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: ▪ The In-Sight vision system is Online. ▪ The *Trigger Enable* bit is set. ▪ The AcquireImage function's Trigger parameter is set to Network, External or Industrial Ethernet.
0	Trigger Enable	This field is set to enable triggering via the Trigger bit. Clear this field to disable the EtherNet/IP triggering mechanism. This bit only affects triggering done via Ethernet/IP, and does not affect any other triggering mode.

Byte 1

Bit	Name	Description
7	Soft Event 7	Allows Spreadsheet soft events to be triggered. Setting any of these bits causes the associated soft event in the Spreadsheet to be triggered.
6	Soft Event 6	
5	Soft Event 5	
4	Soft Event 4	
3	Soft Event 3	
2	Soft Event 2	
1	Soft Event 1	
0	Soft Event 0	

Byte 2-3

Byte	Name	Description
2-3	Job Load ID (4.9.x and earlier)	The ID number (1-999) of the job to load when the Initiate Job Load bit is set by the PLC.

Byte 4-495

Byte	Name	Description
4 - 495	User Data (0 - 491)	This data is sent to the In-Sight Explorer spreadsheet via the ReadEIPBuffer function. The buffer is written exactly as it appears in the PLC, with the bits appearing in the same order as they are defined in RSLogix 5000.

> **Note**: In job deployment environments in which In-Sight Explorer or the VisionView application are monitoring inspections, if the job depends upon a Soft Event (configured as a Timer function, for example) to trigger a spreadsheet event, it may cause the inspection of an image to be delayed if it is triggered shortly before the acquisition cycle completes. If the job file is large (i.e. it contains many Vision Tools, such as Pattern Match, Flaw Detection and/or InspectEdge tools, in addition to additional job logic), the update required by In-Sight Explorer or VisionView may prevent an image from being inspected until the display update is queued. For applications which require exact timing (e.g. measured in the 10s of milliseconds), this update may result in a delay in determining a pass/fail results, and thus the transmission of that result to the next station (e.g. a PLC or motion controller) in the inspection process. Therefore, in these types of application environments, Cognex recommends that Soft Events not be utilized to avoid the possibility of delayed inspections.

I/O Assembly Data Attribute Format - In-Sight Firmware Version 4.1.0 - 4.7.4

Input Assemblies - Instance 11

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11	0	Online	Offline Reason			Missed Acq	Acquiring	Trigger Ack	Trigger Ready
	1	Reserved	Job Load Failed	Job Load Completed	Job Loading	Results Valid	Results Buffer Overrun	Inspection Completed	Inspecting
	2	Acquisition ID (16-bit integer)							
	3								
	4	Inspection ID (16-bit integer)							
	5								
	6	Inspection Result (16-bit integer)							
	7								
	8	Inspection Results 0							
	...								
	499	Inspection Results 491							

Byte 0

<table>
<tr><th>Bit</th><th>Name</th><th>Description</th></tr>
<tr><td>7</td><td>Online</td><td>This bit is set when the In-Sight vision system is Online, and cleared when the vision system is Offline. When the vision system is Offline, examine the <i>Offline Reason</i> field to determine the reason.</td></tr>
<tr><td>6 - 4</td><td>Offline Reason</td><td>This field is a 3-bit field used to identify the cause of why an In-Sight vision system is Offline:
<table>
<tr><th>Offline Reason</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Online</td><td>The vision system is Online.</td></tr>
<tr><td>1</td><td>Programming</td><td>The vision system's job is being modified.</td></tr>
<tr><td>2</td><td>Discrete Offline</td><td>A discrete signal is holding the vision system Offline.</td></tr>
<tr><td>3</td><td>Comm. Offline</td><td>A communications protocol is holding the vision system Offline.</td></tr>
</table>
<b>Note</b>: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.</td></tr>
<tr><td>3</td><td>Missed Acq</td><td>Set when an In-Sight vision system misses an acquisition trigger, regardless how the acquisition was triggered; cleared when an acquisition is successfully triggered.</td></tr>
<tr><td>2</td><td>Acquiring</td><td>Set when the In-Sight vision system is currently acquiring an image; either by setting the Trigger bit or by an external hardware trigger.</td></tr>
</table>

Bit	Name	Description
1	Trigger Ack	Indicates when the In-Sight vision system has been triggered by setting the Trigger bit; this bit will stay set until the Trigger bit is cleared.
0	Trigger Ready	Indicates when the In-Sight vision system can accept a new trigger. This bit is high when the vision system is Online, the Trigger Enable bit is set and the vision system is not currently acquiring an image.

Byte 1

Bit	Name	Description
7	Reserved	Unused.
6	Job Load Failed	This bit is set when the last job load attempt failed. It is cleared the next time a job is successfully loaded. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
5	Job Load Completed	This bit is toggled upon the completion of a job load operation. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
4	Job Loading	This bit is set when loading a new job. **Note**: This bit only functions when the job load was initiated by the PLC using EtherNet/IP.
3	Results Valid	Set when the Inspection ID, Inspection Result and Inspection Results fields are valid. **Note**: Use this bit to indicate that inspection results are ready to be read.
2	Results Buffer Overrun	This field is set when the Buffer Results Enable bit is set and the In-Sight vision system has discarded a set of inspection results because the PLC has not acknowledged the results by setting the Inspection Results Ack bit.
1	Inspection Completed	This bit is toggled on the completion of an inspection. **Note**: Use this bit to indicate that results are ready to read. Use ResultsValid to determine when to read data.
0	Inspecting	This bit is set when the In-Sight vision system is executing spreadsheet functions.

Byte 2-3

Name	Description
Acquisition ID (16-bit Integer)	This ID increments on the completion of every acquisition regardless of the trigger source, and can be used to synchronize an Acquisition with its Inspection Results.

Byte 4-5

Name	Description
Inspection ID (16-bit Integer)	The acquisition ID associated with this set of results. This value increments once each inspection cycle at the arrival of new results.

Byte 6-7

Name	Description
Inspection Result (16-bit Integer)	The pass/fail code that is associated with this set of results. This value cannot be set and is always 0.

Byte 8-499

Name	Description
Inspection Results (0 - 491)	The inspection results set by the WriteEIPBuffer function in the In-Sight Explorer spreadsheet or Ethernet/IP Communications tab in EasyBuilder. The buffer is written exactly as it appears in the In-Sight spreadsheet, with the bits appearing in the same order as they are defined in the

Name	Description
	FormatOutputBuffer function (Spreadsheet) or Format Outputs tab (EasyBuilder).

Output Assemblies - Instance 21

Instance	Byte	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
21	0	Force Offline	Reserved	Reserved	Initiate Job Load	Inspection Results Ack	Buffer Results Enable	Trigger	Trigger Enable
	1	Soft Event 7	Soft Event 6	Soft Event 5	Soft Event 4	Soft Event 3	Soft Event 2	Soft Event 1	Soft Event 0
	2	Job Load ID							
	3								
	4	User Data 0							
	...								
	495	User Data 491							

Byte 0

Bit	Name	Description
7	Force Offline	When this bit is set, the In-Sight vision system is taken Offline until the bit is cleared again.
6 - 5	Reserved	Unused.
4	Initiate Job Load (4.9.x and earlier)	When set, the vision system loads the job ID specified in the Job Load ID field.
3	Inspection Results Ack	When the Buffer Results Enable bit is set, the Inspection Results Ack bit acknowledges that the PLC has received the Inspection ID, Inspection Result and Inspection Results data.
2	Buffer Results Enable	Set this bit to hold the Inspection ID, Inspection Result and Inspection Results fields constant while the ResultsValid bit is set. Set the Inspection Results Ack bit to acknowledge them. The In-Sight vision system responds to the acknowledgement by clearing the Results Valid bit. Once the Inspection Results Ack field is cleared, Results Valid remains cleared until new results are sent to the PLC.
1	Trigger	Setting this bit triggers an acquisition when the following conditions are met: ▪ The In-Sight vision system is Online. ▪ The *Trigger Enable* bit is set. ▪ The AcquireImage function's Trigger parameter is set to Network, External or Industrial Ethernet.
0	Trigger Enable	This field is set to enable triggering via the Trigger bit. Clear this field to disable the EtherNet/IP triggering mechanism. This bit only affects triggering done via Ethernet/IP, and does not affect any other triggering mode.

Byte 1

Bit	Name	Description
7	Soft Event 7	Allows Spreadsheet soft events to be triggered. Setting any of these bits causes the associated soft event in the Spreadsheet to be triggered.
6	Soft Event 6	

Bit	Name	Description
5	Soft Event 5	
4	Soft Event 4	
3	Soft Event 3	
2	Soft Event 2	
1	Soft Event 1	
0	Soft Event 0	

Byte 2-3

Byte	Name	Description
2-3	Job Load ID (4.9.x and earlier)	The ID number (1-999) of the job to load when the Initiate Job Load bit is set by the PLC.

Byte 4-495

Byte	Name	Description
4 - 495	User Data (0 - 491)	This data is sent to the In-Sight Explorer spreadsheet via the ReadEIPBuffer function. The buffer is written exactly as it appears in the PLC, with the bits appearing in the same order as they are defined in RSLogix 5000.

> **Note**: In job deployment environments in which In-Sight Explorer or the VisionView application are monitoring inspections, if the job depends upon a Soft Event (configured as a Timer function, for example) to trigger a spreadsheet event, it may cause the inspection of an image to be delayed if it is triggered shortly before the acquisition cycle completes. If the job file is large (i.e. it contains many Vision Tools, such as Pattern Match, Flaw Detection and/or InspectEdge tools, in addition to additional job logic), the update required by In-Sight Explorer or VisionView may prevent an image from being inspected until the display update is queued. For applications which require exact timing (e.g. measured in the 10s of milliseconds), this update may result in a delay in determining a pass/fail results, and thus the transmission of that result to the next station (e.g. a PLC or motion controller) in the inspection process. Therefore, in these types of application environments, Cognex recommends that Soft Events not be utilized to avoid the possibility of delayed inspections.

Mapping of I/O Assembly Data Attribute Components

The following table indicates the I/O Assembly Data attribute mapping for In-Sight vision systems:

Data Component Name	Class		Instance Number	Attribute	
	Name	Number		Name	Number
Set Offline	Vision	78 Hex	1	ForceOffline	8
Trigger Enable	Vision	78 Hex	1	AcqTriggerEnable	9
Trigger	Vision	78 Hex	1	AcqTrigger	10
BufferResultsEnable	Vision	78 Hex	1	BufferResultsEnable	13
Soft Event N	Vision	78 Hex	1	SoftTrigger bit N	18
Online	Vision	78 Hex	1	Online	6
OfflineReason	Vision	78 Hex	1	OfflineReason	7
Trigger Ready Trigger Ack Exposure Complete Missed Acquisition	Vision	78 Hex	1	AcqStatusRegister	11

Data Component Name	Class		Instance Number	Attribute	
	Name	Number		Name	Number
User Data	Vision	78 Hex	1	User Data	12
System Busy Inspection Completed Results Buffer Overrun Results Valid	Vision	78 Hex	1	InspectionStatusRegister	14
Inspection ID Inspection Data	Vision	78 Hex	1	InspectionResults	16

Vision Object Attributes

This section describes the various attributes and services that exist in the EtherNet/IP Vision Object.

Class Code: 78 Hex - This object models all operations available to the In-Sight vision system running In-Sight 4.x.x and earlier firmware.

Instance Attributes

Attribute ID	Access Rule	Name	Data Type	Description of Attribute
6	Get	Online	BOOL	0 = Offline 1 = Online
7	Get	OfflineReason	BYTE	0 = Online 1 = Programming 2 = Discrete Offline 3 = Comm. Offline 4-255 = Reserved
8	Set	ForceOffline	BOOL	1 = Force the In-Sight vision system Offline.
9	Set	AcqTriggerEnable	BOOL	0 = Acquisition Trigger is disabled. Trigger Ack is reset to 0. 1 = Acquisition Trigger is enabled.
10	Set	AcqTrigger	BOOL	When AcqTriggerEnable is True, the In-Sight vision system will acquire an image when AcqTrigger changes from 0 to 1.
11	Get	AcqStatusRegister	BYTE	Bit 0: Trigger Ready Bit 1: Trigger Ack Bit 2: Acquiring Bit 3: Missed Acquisition Bit 4-7: Reserved
12	Set	UserData	ARRAY of BYTE	User defined data that may be used as an input to the inspection.
13	Set	BufferResultsEnable	BOOL	When BufferResultsEnable is True, it enables buffering of inspection results until the ResultsAck attribute is set to True.
14	Get	InspectionStatusRegister	BYTE	Bit 0: Inspecting Bit 1: Inspection Completed Bit 2: Results Buffer Overrun Bit 3: Results Valid Bit 4: Job Loading Bit 5: Job Load Completed Bit 6: Job Load Failed Bit 7: Reserved

Attribute ID	Access Rule	Name	Data Type	Description of Attribute
15	Set	InspectionResultsAck	BOOL	Acknowledges that the client received the inspection results.
16	Get	InspectionResults	STRUCT of	The last inspection results.
		InspectionID	UINT	Inspection counter.
		Reserved	UINT	Reserved field.
		InspectionResults	ARRAY of BYTE	Results from the last inspection.
17	Get	InspectionResultString	STRING	User-defined result string.
18	Set	SoftTrigger	BYTE	Bit 0: Soft Trigger 0 Bit 1: Soft Trigger 1 Bit 2: Soft Trigger 2 Bit 3: Soft Trigger 3 Bit 4: Soft Trigger 4 Bit 5: Soft Trigger 5 Bit 6: Soft Trigger 6 Bit 7: Soft Trigger 7
19	Set	JobID	INT	The ID number of the currently loaded job, or -1 if the current job does not have an ID. Setting this value (between 1 and 154) will load the job with that associated ID.
20	Set	JobName	STRING	The file name of the currently loaded job, or an empty string if the file name is unknown. Setting this value will load the job with that associated file name.

Online Attributes

The Online Attributes indicate whether an In-Sight vision system is in an Online or Offline state. When the In-Sight vision system is Offline, the OfflineReason attribute can be used to determine the reason as to why the In-Sight vision system is Offline.

OfflineReason Attribute Values

Offline Reason	Name	Description
0	Online	The vision system is Online.
1	Programming	The vision system's job is being modified.
2	Discrete Offline	A discrete signal is holding the vision system Offline.
3	Comm. Offline	A communications protocol is holding the vision system Offline.

> **Note**: It is possible to have multiple devices holding the In-Sight vision system Offline. In this scenario, this field will return the channel with the lowest reason code.

The ForceOffline attribute can be used to force the In-Sight vision system into an Offline state. When the ForceOffline attribute is set to True, the OfflineReason attribute will be set to Comm. Offline, if the In-Sight vision system is being held Offline for no other reason.

Acquisition Attributes

> **Note**: If the In-Sight vision system will be configured to accept an acquisition trigger from a PLC/Motion Controller via a Native Mode command, Cognex recommends that the SetEvent and Wait function be utilized, with the Event code set to 8 (**SW8**). This will ensure that vision system waits for both the acquisition and inspection to be completed before sending a "complete" response back to the

PLC/Motion Controller, and that previous inspection results are not being sent to the PLC/Motion Controller. The "complete" response from the vision system can also then be used to create conditional PLC logic that sends a read request for the inspection results.

The Vision Object can be triggered to acquire images by using the Acquisition Attributes. In order to reset the Acquisition Attributes in the Vision Object, the AcqTriggerEnable attribute must be set to False, until the AcqStatusRegister is 0. Then, the AcqTriggerEnable attribute can be set to True to enable acquisition via the Vision Object. When the Acquisition Trigger is ready to accept triggers, the Trigger Ready bit in the AcqStatusRegister will be set.

While the AcqTriggerEnable attribute is set to True, each time the Vision Object sees the AcqTrigger attribute change from 0 to 1, it will initiate an image acquisition. When setting this via implicit messaging, to guarantee that the change is seen by the Vision Object, the attribute should be held in the changed state until that same value is seen by the Trigger Ack bit of the AcqStatusRegister.

Note: This is not necessary when using explicit messaging, because the change will always be seen.

During an acquisition, the Trigger Ready bit in the AcqStatusRegister will be cleared and the Acquiring bit will be set. When the acquisition is completed, the Acquiring bit will be cleared; and once the acquisition system is ready to begin a new image acquisition, the Trigger Ready bit will again be set.

If other methods of acquisition triggering are employed, the Exposure Complete and Missed Acquisition bits of the AcqStatusRegister will still be active.

Inspection Results Attributes

When an image is acquired as a result of an acquisition trigger, it is placed in a queue for inspection. As the inspection is running on the image, the Inspecting bit of the InspectionStatusRegister is set. Once the inspection is completed, the Inspecting bit is cleared and the Inspection Completed bit is toggled.

The BufferResultsEnable attribute determines how inspection results are handled in the Vision Object. If the BufferResultsEnable attribute is set to False, then the inspection results are immediately placed into the InspectionResults attribute.

If the BufferResultsEnable attribute is set to True, then the previous inspection results remain in the InspectionResults attribute until they are acknowledged by setting the InspectionResultsAck attribute to True. Once the Results Valid bit is cleared, the InspectionResultsAck attribute should be set to False to allow new results to be placed in to the InspectionResults attribute. This can be used to establish a handshake to obtain results between the Vision Object and the client.

Behavior of InspectionStatusRegister

Inspection Status Register Bit	Bit Name	Results If Buffering Is Disabled	Results If Buffering Is Enabled
0	Inspecting	Set when inspecting an image.	Set when inspecting an image.
1	Inspection Completed	Toggled on completion of an inspection.	Toggled on completion of an inspection.
2	Results Buffer Overrun	Always cleared.	Set when inspection results could not be queued because the client has failed to acknowledge a previous result. Cleared when the inspection result is successfully queued.
3	Results Valid	Cleared when an inspection begins; set when an inspection is completed.	Set when new results are placed in the InspectionResults attribute. Stays set until the results are acknowledged by setting the InspectionResultsAck attribute to True.
4	Job Pass*	Set when an inspection is completed and the Job Pass/Fail cell indicates pass; otherwise it remains cleared. **Note**: The Job Pass bit	Set when new results are placed in the InspectionResults attribute and the Job Pass/Fail cell indicated pass for that result set; otherwise it remains cleared. The state will not change until the results are acknowledged by setting the InspectionResultsAck attribute to True, at which time the bit will be cleared. **Note**: The Job Pass bit will be valid prior to the Results Valid bit

Inspection Status Register Bit	Bit Name	Results If Buffering Is Disabled	Results If Buffering Is Enabled
		will be valid prior to toggling the Inspection Completed bit.	being set.
*Job Pass bit is only included in Major Revision 10 and later.			

Job Attributes

The Vision Object can load jobs stored in the non-volatile flash memory of an In-Sight vision system by setting the JobID and JobName attributes. If the load fails, the set operation will also fail.

Vision Object Services

Common Services

The Vision Object provides the following Common Services:

Service Code	Service Name	Description of Service
05 Hex	Reset	Resets the Vision Object
0E Hex	Get_Attribute_Single	Returns the contents of the specified attribute.
10 Hex	Set_Attribute_Single	Modifies an attribute value.

Reset Service

The Reset Service will reset and disable the acquisition triggering system in the Vision Object. It will also disable and clear the inspection results buffering system in the Vision Object.

Object Specific Services

The Vision Object provides the following Object Specific Services:

Service Code	Service Name	Description of Service
32 Hex	Acquire	Triggers a single acquisition.
33 Hex	Reserved	Deprecated.
34 Hex	SendNativeCmd	Sends a Native Mode command to the In-Sight vision system.
35 Hex	GetInspectionResults	Gets the InspectionResults attribute.
36 Hex	Transfer User Data	Transfers data into a portion of the User Data Holding Buffer.
37 Hex	Set User Data	Indicates to the vision system that it should latch the User Data Holding Buffer into the User Data field to give the In-Sight access to the user data.

Acquire Service

The Acquire Service of the Vision Object will cause an acquisition to be triggered, if the acquisition system is ready to acquire an image. If the acquisition could not be triggered, then the Missed Acquisition bit of the AcqStatusRegister will be set until the next successful acquisition.

If the command fails, the Acquire Service will return the vendor specific error code 1F.

SendNativeCmd Service

The SendNativeCmd Service sends any documented Native Mode command string to the In-Sight vision system. The request data for the SendNativeCmd should include the Native Mode command string that is to be sent to the In-Sight vision system; the reply data will contain the string result of the Native Mode command.

SendNativeCmd Request Data Format

Name	Type	Description
Command	STRING	The Native Mode command to process.

SendNativeCmd Response Data Format

Name	Type	Description
Result	STRING	The string result of the Native Mode command.

If the command fails, the service will return one of the following vendor specific error codes:

Error Code	Description	Native Mode Return Code
1	Bad Command	-
4	No Answer - The system is too busy.	
100	Command Failed	0
101	Command Cannot Be Executed	-1
102		-2
103		-3
104		-4
105		-5
106		-6

GetInspectionResults Service

The GetInspectionResults Service (code 0x35) reads data from the InspectionResults attribute of the Vision Object. This service can be used to read data from the InspectionResults attribute from a device which does not support receiving messages of 260 bytes.

GetInspectionResults Request Data Format

Name	Type	Description
Size	UINT	The number of bytes of the InspectionResults attribute to read.
Offset	UINT	(Optional) The offset into the InspectionResults attribute. This specifies the first byte of the InspectionResults attribute to read.

GetInspectionResults Response Data Format

Name	Type	Description
Data	Array of BYTE	The InspectionResults attribute data.

Transfer User Data Command

This command is used to transfer data into a portion of the User Data Holding Buffer. The User Data Holding Buffer will be latched to the User Data field when the Set User Data command is issued.

Request Format

Name	Type	Description

Name	Type	Description
Size	UINT	The number of bytes of the User Data field to write.
Offset	UINT	The offset into the User Data field.
Data	Array of BYTE	User Data to write into the User Data field.

Set User Data Command

This command is used by the PLC to indicate to the vision system that it should latch the User Data Holding Buffer into the User Data field to give the In-Sight access to the user data.

Get Data from an In-Sight Vision System – 4.x.x Spreadsheet

In order to get data from the In-Sight Explorer spreadsheet to a ControlLogix PLC, the data must be pushed into the EtherNet/IP stack by using the WriteEIPBuffer function. This function takes a buffer of data created by the FormatOutputBuffer function and writes it to the data area in the In-Sight vision system's EtherNet/IP Input Assembly (input to the network). This data is then transferred to the PLC during the next RPI cycle.

The EIP buffer writes to this data area are queued, meaning that they do not happen during spreadsheet execution, but as a separate event slightly after the execution of the spreadsheet. This means that there is a small window of time during which an inspection will be complete, but a new buffer write will not have resolved.

It is important to use the *ResultsValid* bit to decide when to read data to prevent subtle intermittent timing issues in your PLC logic that can arise from this setup, as *ResultsValid* will only become high when data are ready to read, and not before.

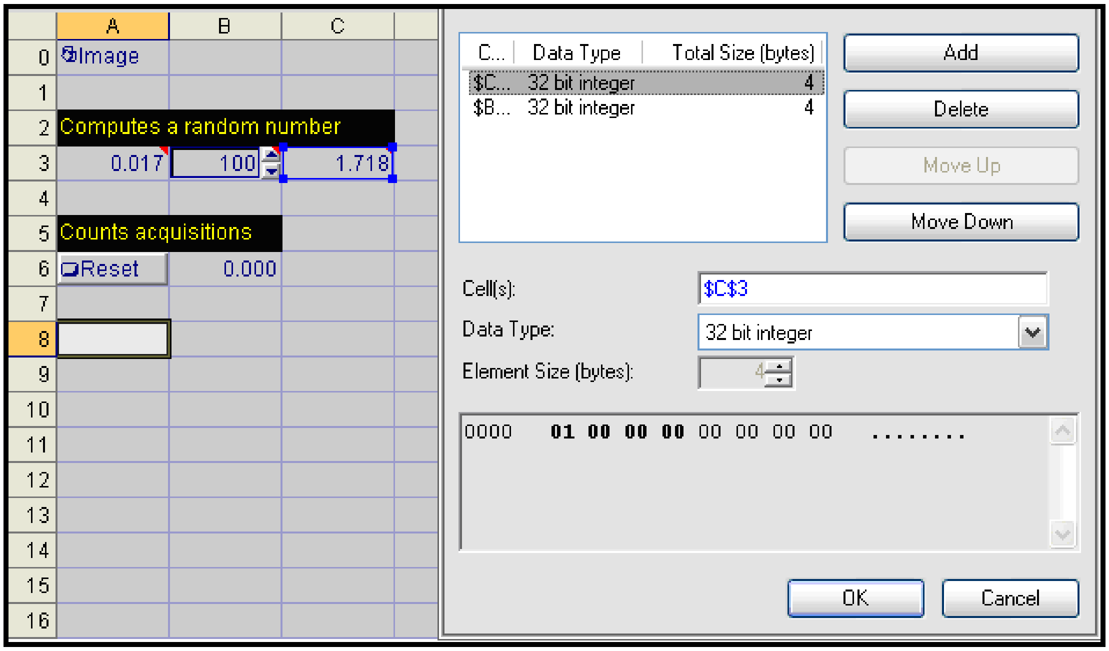
- EXAMPLE: Getting 32 bit integer data from an In-Sight vision system
- EXAMPLE: Getting floating point value data from an In-Sight vision system

---

EXAMPLE: Getting 32 bit integer data from an In-Sight vision system

The following steps explain how to format the data that will be sent from an In-Sight vision system to a ControlLogix PLC.

1. To begin, using In-Sight Explorer, create a new job.
2. From the Palette's Snippets tab, add these two Snippets to the spreadsheet: *Acquisition > AcqCounter* and *Math & Logic > Random*.
3. Open the AcquireImage cell and set the **Trigger** parameter to *Continuous*.
4. Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category and double-click on the FormatOutputBuffer function to insert it into the spreadsheet.
5. From the FormatOutputBuffer dialog, click on the *Add* button. This will initiate the cell selection mode; select the "Scaled random number" cell of the *Random* snippet.
6. From the FormatOutputBuffer dialog, use the *Data Type* pull-down menu to change the *Data Type* to *32 bit integer*.
7. From the FormatOutputBuffer dialog, click on the *Add* button again. This will initiate cell selection mode; select the count cell of the *AcqCounter* snippet.



	A	B	C
0	Image		
1			
2	Computes a random number		
3	0.017	100	1.718
4			
5	Counts acquisitions		
6	Reset	0.000	
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

C...	Data Type	Total Size (bytes)
$C...	32 bit integer	4
$B...	32 bit integer	4

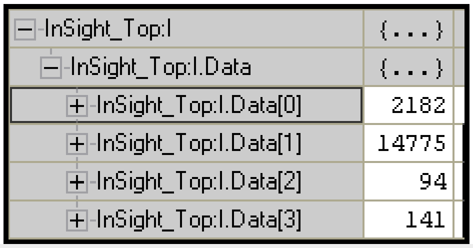
Cell(s): C3

Data Type: 32 bit integer

Element Size (bytes): 4

```
0000    01 00 00 00 00 00 00 00    ........
```

8. Close the [FormatOutputBuffer](#) by clicking the *OK* button.
9. Right-click an empty cell and select [Insert Function](#). From the left-side of the screen, expand the *Input/Output* category, click on the *Network* subcategory and double-click on the [WriteEIPBuffer](#) function to insert it into the spreadsheet.
10. Set the WriteEIPBuffer function's *Buffer* parameter as a [cell reference](#) to the just created FormatOutputBuffer Buffer data structure.
11. Place the In-Sight vision system [Online](#).
12. The data should now be displayed in the *Controller -> Controller Tags* node of RSLogix 5000.



InSight_Top:I	{...}
InSight_Top:I.Data	{...}
InSight_Top:I.Data[0]	2182
InSight_Top:I.Data[1]	14775
InSight_Top:I.Data[2]	94
InSight_Top:I.Data[3]	141

> **Note**: The RSLogix Monitor Tags tab must be selected in order to display the data values.

EXAMPLE: Getting floating point value data from an In-Sight vision system

If a floating point or a mix of data types needs to be sent, in ControlLogix create a User Defined Type. Copy the floating-point data into the newly created User Defined Type from the In-Sight vision system's input tag.

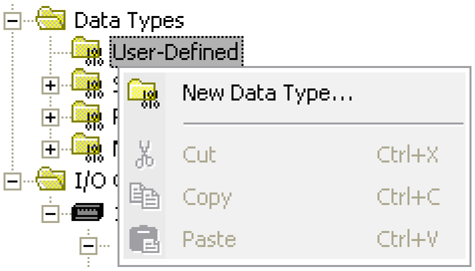
The steps below illustrate how to copy two cells as floating-point values:

1. Place the In-Sight vision system [Offline](#).
2. Open the previously created [FormatOutputBuffer](#) function.
3. From the FormatOutputBuffer dialog, use the *Data Type* pull-down menu to change the *Data Type* from *32 bit integer* to *32 bit floating point*.

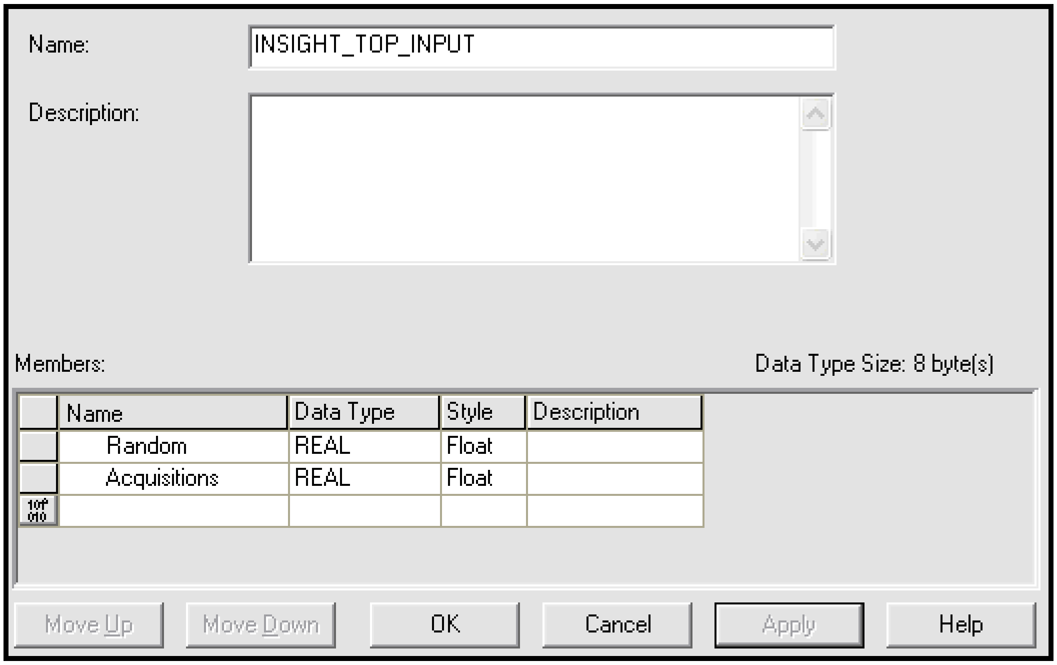
> **Note**: Matching the *Data Type* being used by the vision system is not necessarily required. The bits are copied verbatim from the vision system's output buffers in the order that they appear in FormatOutputBuffer or the Format Outputs tab, regardless of the data type used to receive the buffer in the PLC.

4. Place the In-Sight vision system back Online.

5. In RSLogix, go into Offline mode.
6. Within RSLogix, right-click on the *Data Types* -> *User-Defined* folder and select *New Data Type* from the menu.



7. Give the new data type a descriptive name and add two floating point values to the *Members:* list.



Name: INSIGHT_TOP_INPUT

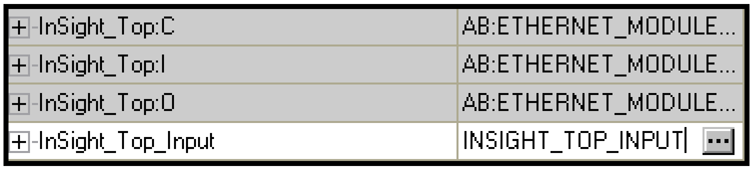
Description:

Members:

Data Type Size: 8 byte(s)

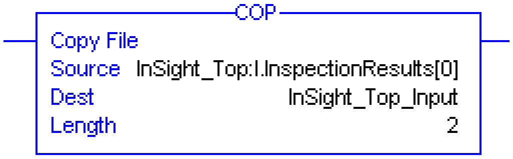
Name	Data Type	Style	Description
Random	REAL	Float	
Acquisitions	REAL	Float	

8. Next, add a new tag to the project's Controller Tags, using the user defined data type.



+InSight_Top:C	AB:ETHERNET_MODULE...
+InSight_Top:I	AB:ETHERNET_MODULE...
+InSight_Top:O	AB:ETHERNET_MODULE...
+InSight_Top_Input	INSIGHT_TOP_INPUT

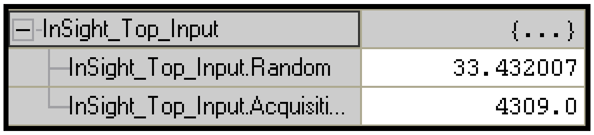
9. Finally, add a COP instruction to the MainRoutine ladder logic program, which will copy the data from the **InSight_Top:I.InspectionResults[0]** tag to the **InSight_Top_Input** tag.



COP
Copy File
Source InSight_Top:I.InspectionResults[0]
Dest InSight_Top_Input
Length 2

> **Note**: The Length field is the number of the input structures that should be copied to the output. In this case, the input structures are 32-bit floating point values, so the COP command copies 64 bits from the vision system's output buffer into the user-defined type in the PLC. If we were using 16-bit integer types (INT)in the PLC's receiving buffers, then we would have used a length of 4 to get the same 64 bits, and if we were using 8-bit integer types (SINT), then we would have used a length of 8 to copy all 64 bits.

10. After this program has been downloaded to the ControlLogix PLC, the data from the Random and Acquisition Counter In-Sight Explorer spreadsheet cells will now appear in the **InSight_Top_Input** tag in the floating point format:



⊟ InSight_Top_Input	{...}
InSight_Top_Input.Random	33.432007
InSight_Top_Input.Acquisiti...	4309.0

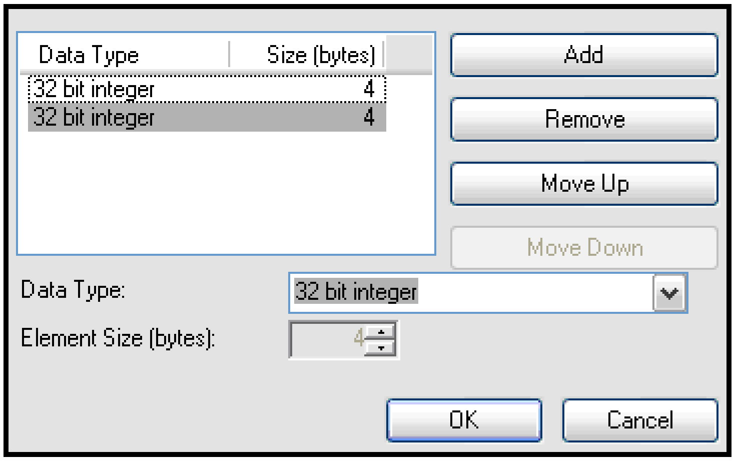
Send Data To an In-Sight Vision System – 4.x.x Spreadsheet

In order to send data from the ControlLogix PLC to the In-Sight Explorer spreadsheet, the data must be pulled from the EtherNet/IP stack by using the ReadEIPBuffer function. This function takes the data format created within the FormatInputBuffer function and reads the data from the data area of the EtherNet/IP *Output Assembly* (output from the network), and formats this data into the In-Sight Explorer spreadsheet. This data is received from the PLC every RPI cycle; the corresponding spreadsheet data is updated only when the spreadsheet executes.

EXAMPLE: Sending 32 bit integer data to an In-Sight Vision System

For this example, create a new job within In-Sight Explorer, then perform the following steps to configure the data that will be received by the ControlLogix PLC.

1. Open the AcquireImage cell and set the **Trigger** parameter to *Continuous*.
2. Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category and double-click on the FormatInputBuffer function to insert it into the spreadsheet.
3. From the FormatInputBuffer dialog, click on the **Add** button and add two 32-bit integers to the list.

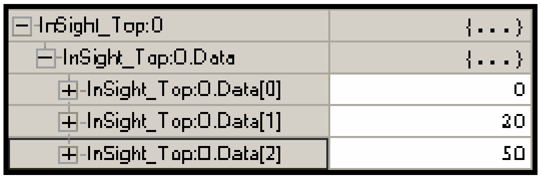


Data Type	Size (bytes)
32 bit integer	4
32 bit integer	4

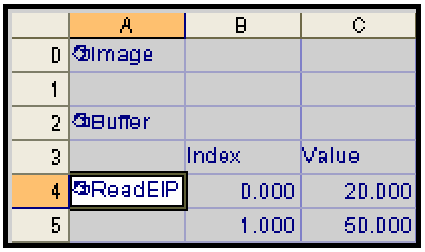
Data Type: 32 bit integer

Element Size (bytes): 4

4. Close the FormatInputBuffer by clicking the *OK* button.
5. Right-click an empty cell and select Insert Function. From the left-side of the screen, expand the *Input/Output* category, click on the *Network* subcategory and double-click on the ReadEIPBuffer function to insert it into the spreadsheet.
6. Set the ReadEIPBuffer function's *Buffer* parameter as a cell reference to the just created FormatOutputBuffer Buffer data structure.
7. The Data Access functions will automatically be added to the spreadsheet based on the fields added to the FormatOutputBuffer function.
8. Place the In-Sight vision system in Online mode.
9. Within RSLogix 5000, open the *Controller Tags* dialog and change the value of the **InSight_Top:O.Data[1]** and **InSight_Top:O.Data[2]** tags. The values in the In-Sight Explorer spreadsheet should change, as well:



InSight_Top:O	{...}
InSight_Top:O.Data	{...}
InSight_Top:O.Data[0]	0
InSight_Top:O.Data[1]	20
InSight_Top:O.Data[2]	50

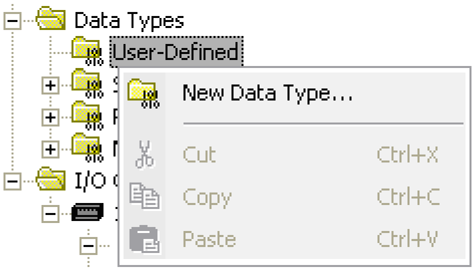


	A	B	C
0	Image		
1			
2	Buffer		
3		Index	Value
4	ReadEIP	0.000	20.000
5		1.000	50.000

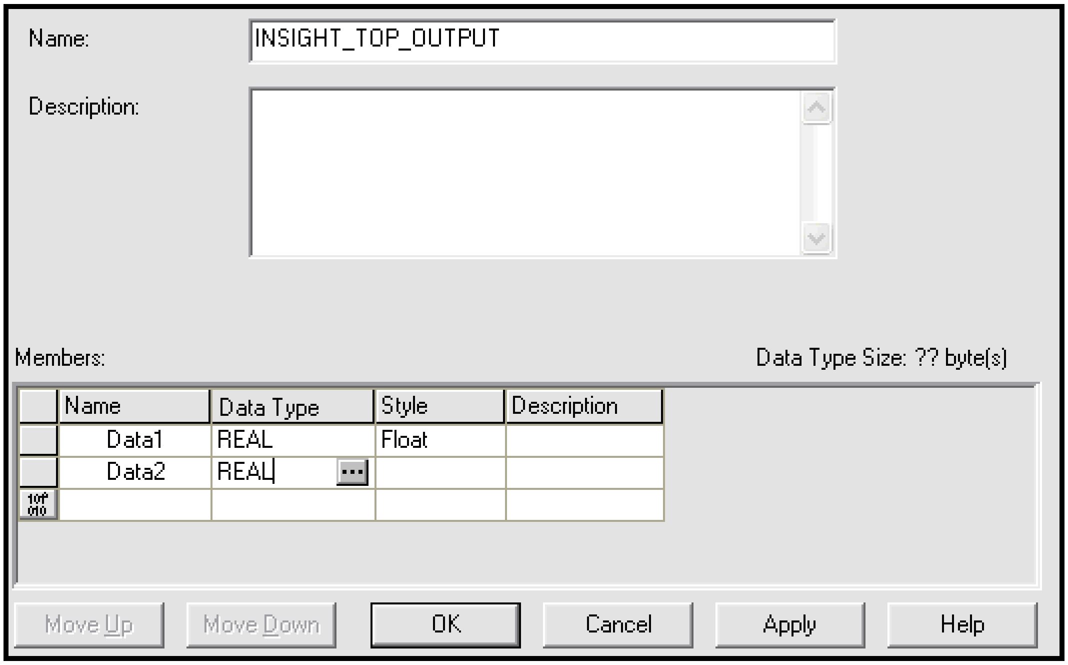
EXAMPLE: Sending floating point value data to an In-Sight vision system

If another data type, for instance a type other than the 32-bit integers type, or a mix of data types needs to be sent, the simplest method to accomplish this is to create a *User Defined Type* in ControlLogix, and then to copy that data from the In-Sight vision system's output tag to the newly created *User Defined Type*. The steps below illustrate how to copy to two cells as floating point values:

1. In RSLogix, go into Offline mode.
2. Within RSLogix, right-click on the *User-Defined ->Data Types* folder and select *New Data Type...* from the menu.



3. Give the new data type a descriptive name and add two floating point values to the *Members:* list.



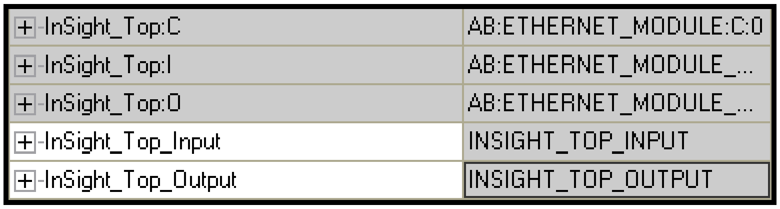
Name: INSIGHT_TOP_OUTPUT

Description:

Members: Data Type Size: ?? byte(s)

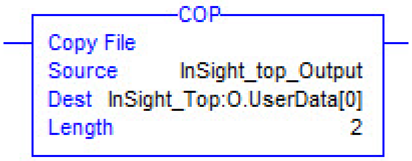
Name	Data Type	Style	Description
Data1	REAL	Float	
Data2	REAL		

4. Next, add a new tag to the project's Controller Tags, using the user defined data type, as shown in the example below.



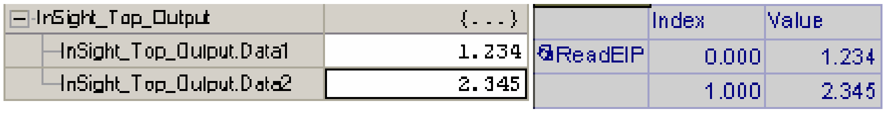
InSight_Top:C	AB:ETHERNET_MODULE:C:0
InSight_Top:I	AB:ETHERNET_MODULE_...
InSight_Top:O	AB:ETHERNET_MODULE_...
InSight_Top_Input	INSIGHT_TOP_INPUT
InSight_Top_Output	INSIGHT_TOP_OUTPUT

5. Finally, add a COP instruction to the MainRoutine ladder logic program, which will copy the data from the **InSight_Top_Output** tag to the **InSight_Top:O.Data[1]** tag.



COP
Copy File
Source InSight_top_Output
Dest InSight_Top:O.UserData[0]
Length 2

6. Place the In-Sight vision system Offline.
7. Open the previously created FormatInputBuffer function, and change the *Data Type* values of both to *32 bit floating point*.
8. Place the In-Sight vision system back Online.
9. After this program has been downloaded to the ControlLogix PLC, the data in the **InSight_Top_Output** tag will now appear in the floating point format:



⊟ InSight_Top_Output	{...}
InSight_Top_Output.Data1	1.234
InSight_Top_Output.Data2	2.345

	Index	Value
ReadEIP	0.000	1.234
	1.000	2.345

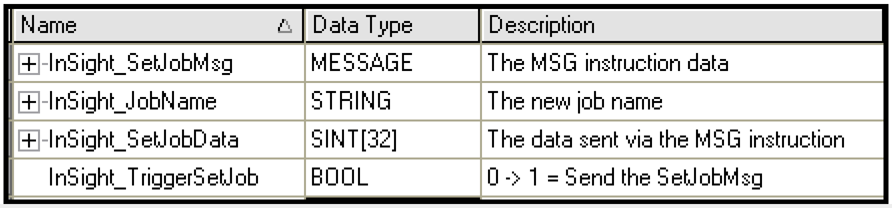
Explicit Messages – 4.x.x

Unlike implicit messages, explicit messages are sent to a specific device and that device always responds with a reply to that message. As a result, explicit messages are better suited for operations that occur less frequently. Explicit messages can be used to read and write the Attributes in the EtherNet/IP Vision Object of the In-Sight Vision System, which may be used for changing jobs, acquiring images, sending Native Mode commands and retrieving result data.

Example: Change a Job

The most common explicit message sent to an In-Sight vision system from a ControlLogix PLC is an instruction to change a job. Explicit messages are sent from a ControlLogix PLC using MSG instructions. The following steps show how to add a MSG instruction in RSLogix to change a job on an In-Sight vision system:

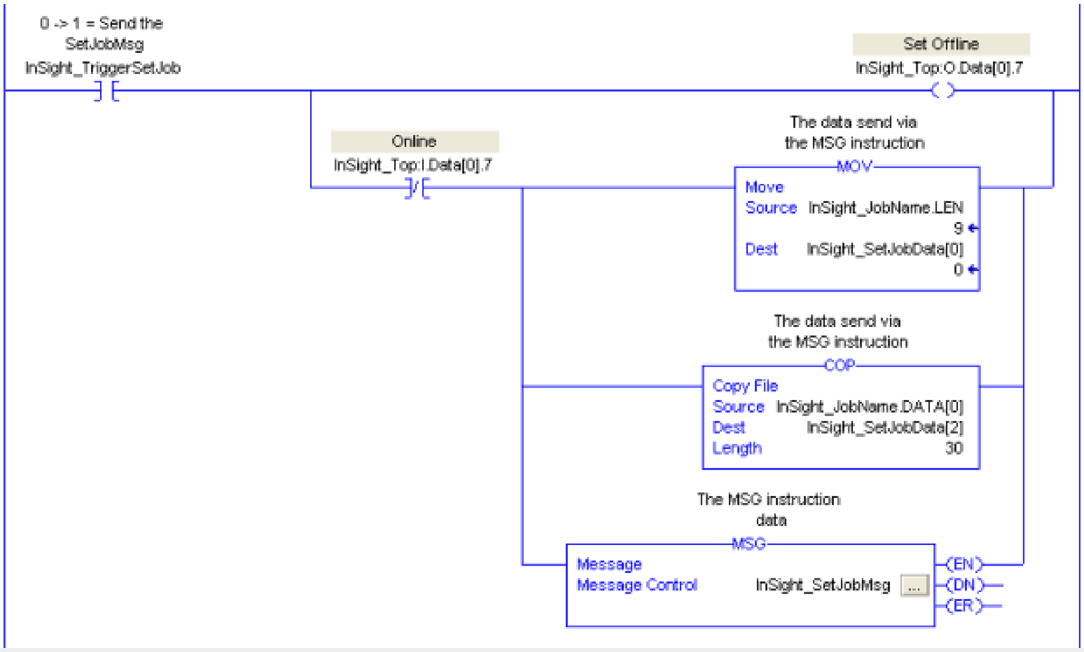
1. Add the following tags to the ControlLogix **Controller Tags** dialog:



Name △	Data Type	Description
⊞-InSight_SetJobMsg	MESSAGE	The MSG instruction data
⊞-InSight_JobName	STRING	The new job name
⊞-InSight_SetJobData	SINT[32]	The data sent via the MSG instruction
InSight_TriggerSetJob	BOOL	0 -> 1 = Send the SetJobMsg

> **Note**: This example assumes a maximum length of 30 characters for the job name. If your job's name is longer, then you will need to extend the SetJobData buffer to include a number of SINT structures equal to the number of characters in your longest job name plus two.

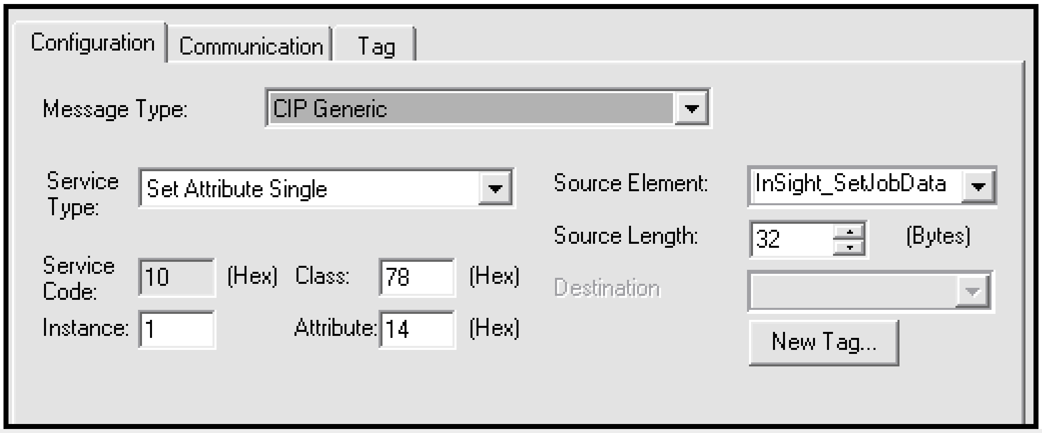
2. Create the following rung in your RSLogix 5000 project:


0 -> 1 = Send the SetJobMsg
InSight_TriggerSetJob
Set Offline
InSight_Top:O.Data[0].7
Online
InSight_Top:I.Data[0].7
The data send via the MSG instruction
MOV
Move
Source InSight_JobName.LEN
9
Dest InSight_SetJobData[0]
0
The data send via the MSG instruction
COP
Copy File
Source InSight_JobName.DATA[0]
Dest InSight_SetJobData[2]
Length 30
The MSG instruction data
MSG
Message
Message Control InSight_SetJobMsg ...
EN
DN
ER

> **Note**: This snippet is copying the length of the job name into the first two bytes of the SetJobData buffer, the string data into the same buffer starting at the third byte, and then sends that buffer to the JobName attribute of the vision system. Make sure to add the ".job" file extension onto the end of the job name.

This rung uses the *Set Offline* bit in the implicit connection to force the In-Sight vision system Offline, because the *JobName* attribute of the *Vision Object* requires the In-Sight vision system to be Offline before it will change the job. The *Set Offline* bit waits for the In-Sight vision system to bring the Online bit low, then sets up the data containing the new job name and sends the MSG instruction to the In-Sight vision system. After the job change is completed, the falling edge of the *TriggerSetJob* tag will cause the In-Sight vision system to go back Online.

3. To setup the MSG instruction, click on the *InSight_SetJobMsg …* button. This will cause the **Message Configuration** dialog to appear:



Message Type: CIP Generic

Service Type: Set Attribute Single

Service Code: 10 (Hex)

Class: 78 (Hex)

Instance: 1

Attribute: 14 (Hex)

Source Element: InSight_SetJobData

Source Length: 32 (Bytes)

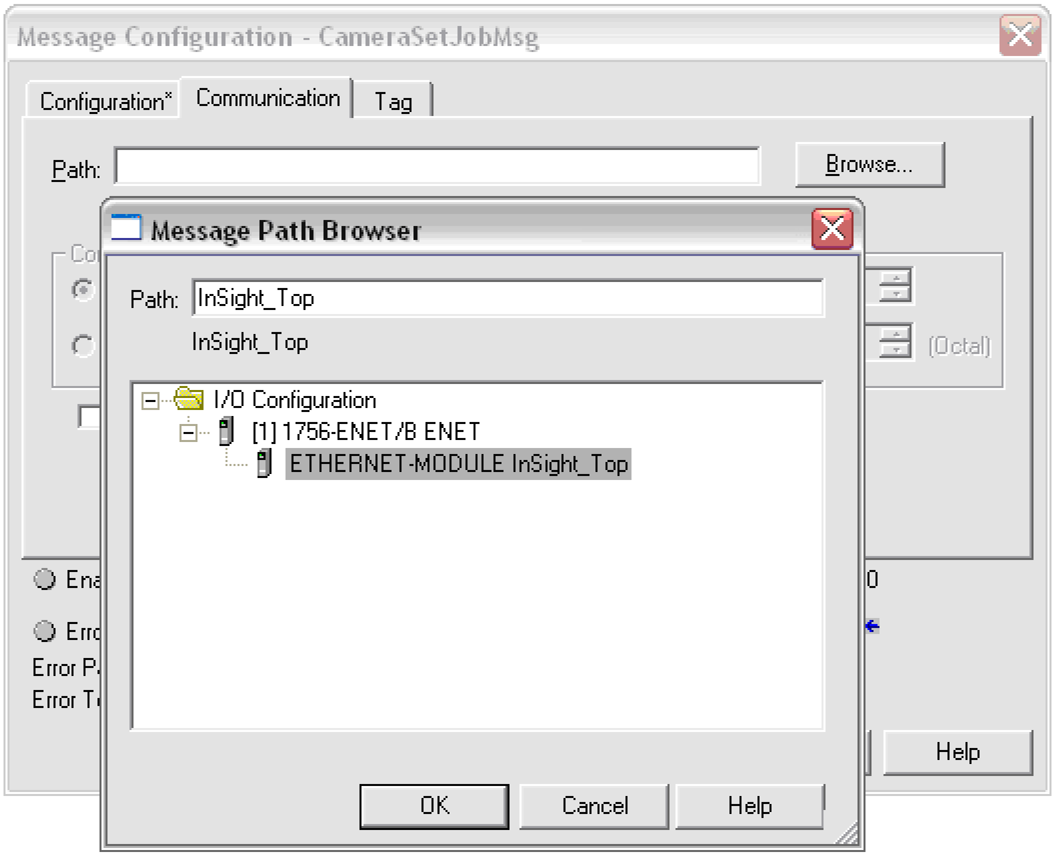
Destination

> **Note**: This example assumes a maximum length of 30 characters for the job name. If your job's name is longer, then you will need to input a Source Length equal to the number of characters in your longest job name plus two.

The following fields need to be configured:

- **Message Type:** This is the type of message that will be sent; choose *CIP Generic* to send a message to vendor specific objects like the *Vision Object*.
- **Service Type:** This is the service code that will be sent to the object; select *Set Attribute Single* to change a single attribute of the *Vision Object*.
- **Service Code:** This field allows any type of service to the object; this field is disabled if a predefined service type is selected.
- **Class:** This is the identifier of the class that the message will be sent to; the ID of the *Vision Object* is *Hex 78*.
- **Instance:** This field specifies the instance number of the object that the message will be sent to; for In-Sight vision systems, the *Vision Object* only has one instance, so this field should be set to 1.
- **Attribute:** The attribute number that the message will be sent to; in this case, the *JobName* attribute has the ID of 14 Hex.
- **Source Element:** This field indicates the source for the data that is being sent with the message. For the *JobName* attribute, a *String* with a 2 byte length header needs to be sent. This string was formatted earlier using the COP and MOV instructions in the run that was created in the previous steps. This field should be set to **InSight_SetJobData** to send the new job name to the *JobName* attribute.
- **Source Length:** This field indicates the number of bytes of the source element that will be sent to the object. In this instance, because all of the data in the source element needs to be sent, the bytes should be set to 32.

4. The MSG instruction now needs to be told which device should have the explicit message sent to it. Click on the *Communication* tab of the **Message Configuration** dialog and press the *Browse...* button. Choose the In-Sight vision system from the I/O configuration as shown below:

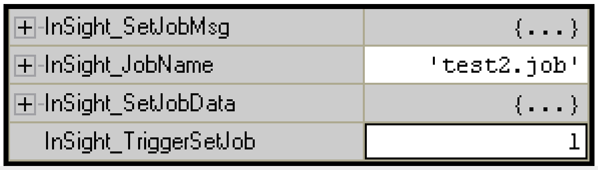


Path: InSight_Top

InSight_Top

- I/O Configuration
  - [1] 1756-ENET/B ENET
    - ETHERNET-MODULE InSight_Top

5. After downloading the program to the ControlLogix PLC, the current job should be able to be changed with RSLogix by using the **InSight_JobName** and the **InSight_TriggerSetJob** controller tags:



+ InSight_SetJobMsg	{...}
+ InSight_JobName	'test2.job'
+ InSight_SetJobData	{...}
InSight_TriggerSetJob	1

> **Note**: Make sure to add the ".job" extension at the end of the job name.

6. The *Job Loading*, *Job Load Complete*, and *Job Load Failed* bits can be used to determine if the job loaded successfully.

Establish the Connection Using the EDS Generated Profile

In-Sight Explorer includes .L5X files, which can be used to copy PLC controller tag data to/from user-friendly program tags, representing signals in the In-Sight vision system running In-Sight 4.10.x or 5.x.x firmware.

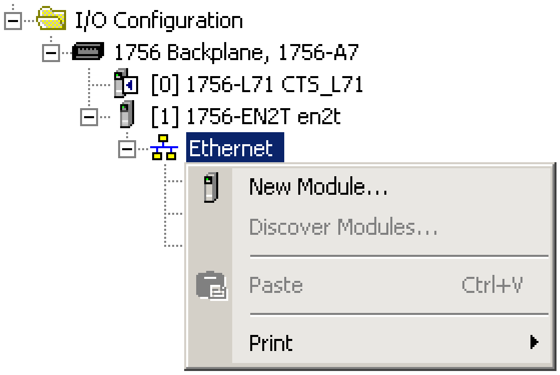
> **Notes**:
>
> - If using an EDS generated profile, although the maximum size of the Input Assembly is 500 bytes, the EDS generated profile will only allow a connection of up to 496 bytes.
> - When using Rockwell RSLogix Studio 5000, version 21 through 25, the In-Sight EDS generated profile's default Input and Output Assembly sizes (496 bytes for each) cannot be edited and will default to the largest Input and Output Assembly sizes.
> - When using Rockwell RSLogix Studio 5000 version 19 and earlier, an Allen-Bradley Generic ETHERNET-MODULE must be used. Reference the charts in the "Integration with RSLogix" section of this document for install guidance.

6) Install the EDS files included with In-Sight Explorer software, if not already installed.

7) Open RSLogix 5000 and load the PLC's project.

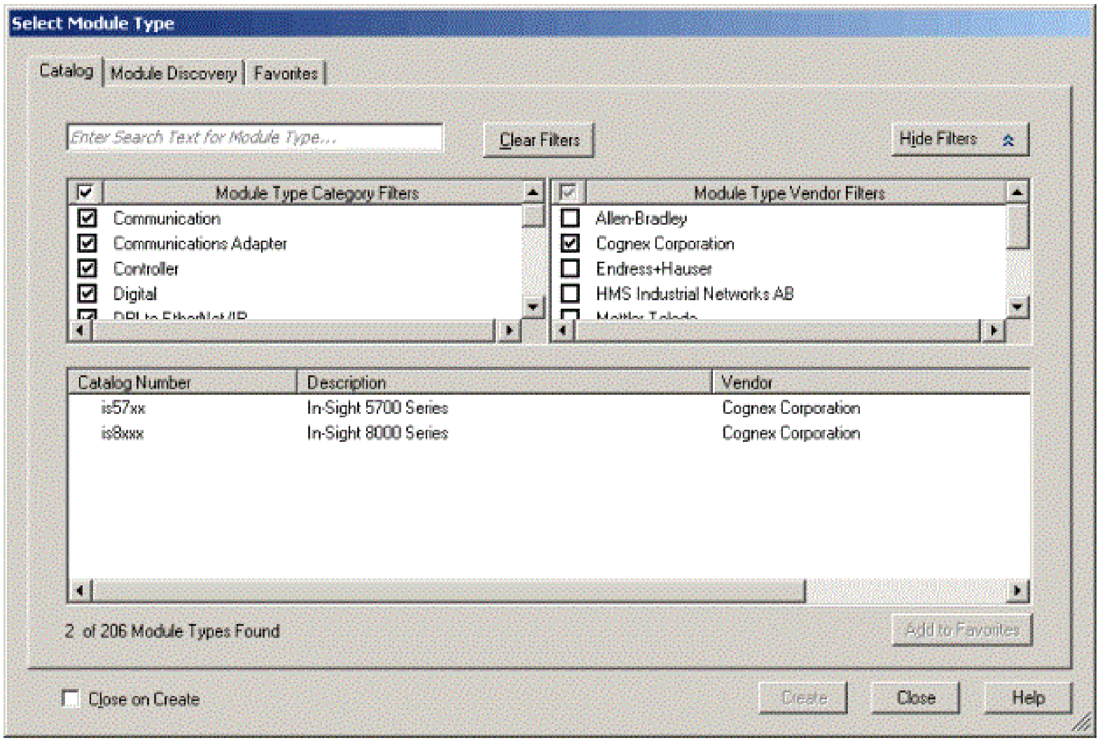
> **Note**: The PLC must be Offline to add connections in RSLogix 5000.

8) Add a connection to your Ethernet Communications Card. Under the I/O Configuration node, select the **Ethernet Node** under the Ethernet Module, right-click on the icon and select **New Module** from the menu.



9) From the Select Module Type dialog, select the type of In-Sight vision system that will be connected to the PLC and click **Create**.

> **Note**: After loading the new EDS files, new module types will show up and the appropriate module should be selected.



☑	Module Type Category Filters
☑	Communication
☑	Communications Adapter
☑	Controller
☑	Digital

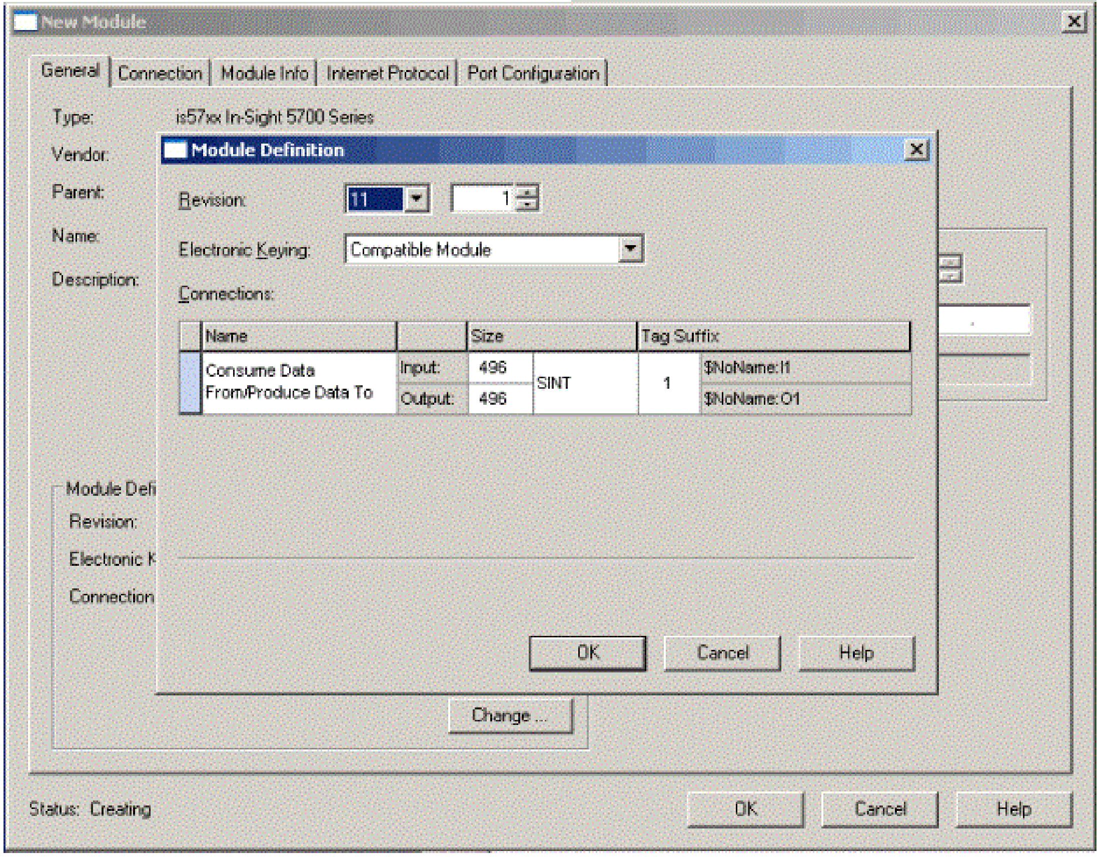
☑	Module Type Vendor Filters
☐	Allen-Bradley
☑	Cognex Corporation
☐	Endress+Hauser
☐	HMS Industrial Networks AB

Catalog Number	Description	Vendor
is57xx	In-Sight 5700 Series	Cognex Corporation
is8xxx	In-Sight 8000 Series	Cognex Corporation

2 of 206 Module Types Found

☐ Close on Create

10) From the New Module dialog, enter the In-Sight vision system's name and IP address, and then click the **Change...** button to configure the size of the connection.



Type: is57xx In-Sight 5700 Series

Revision: 11 . 1

Electronic Keying: Compatible Module

Connections:

Name		Size		Tag Suffix
Consume Data From/Produce Data To	Input:	496	SINT	1	$NoName:I1
	Output:	496			$NoName:O1

Status: Creating

11) After adding the vision system to the RSLogix project, the L5X rung import file can be used to create structured data for the connection. Within the main program, right-click on an empty rung and select the **Import Rungs...** option. Navigate to the .L5X file and click the **Import...** button.

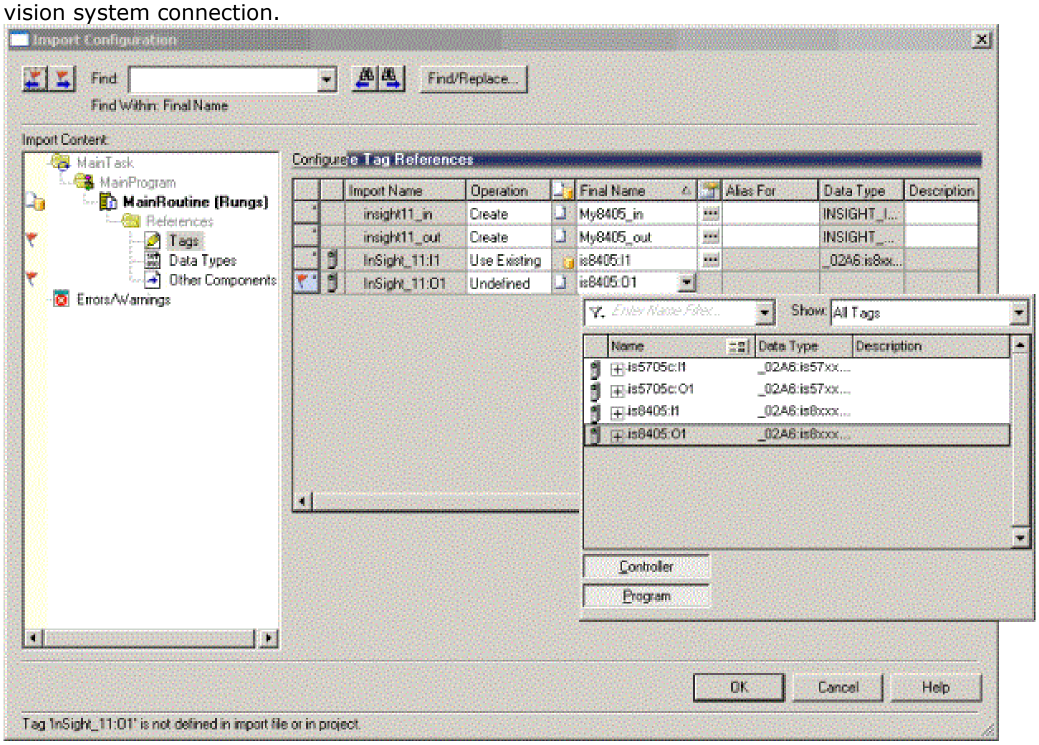
    - If using In-Sight 4.10.x firmware, import the **InSight_12_CopyRung.L5X** file, which is installed to the following location:
      C:\Program Files (x86)\Cognex\In-Sight\In-Sight Explorer 5.x.x\Factory Protocol Description\RSLogix

    - If using In-Sight 5.x.x firmware, import the **InSight_11_CopyRung.L5X** file, which is installed to the following location:
      C:\Program Files (x86)\Cognex\In-Sight\In-Sight Explorer 5.x.x\Factory Protocol Description\RSLogix

7. In the Import Configuration dialog, select **Tags** and change the names of the created tags to match the configuration.

    - If using In-Sight 4.10.x firmware, set the **InSight_12:I** and **InSight_12:O** tags to the connection controller tags in the project so that they match the name given to the In-Sight vision system connection.

    - If using In-Sight 5.x.x firmware, set the **InSight_11:I1** and **InSight_11:O1** tags to the connection controller tags in the project so that they match the name given to the In-Sight vision system connection.

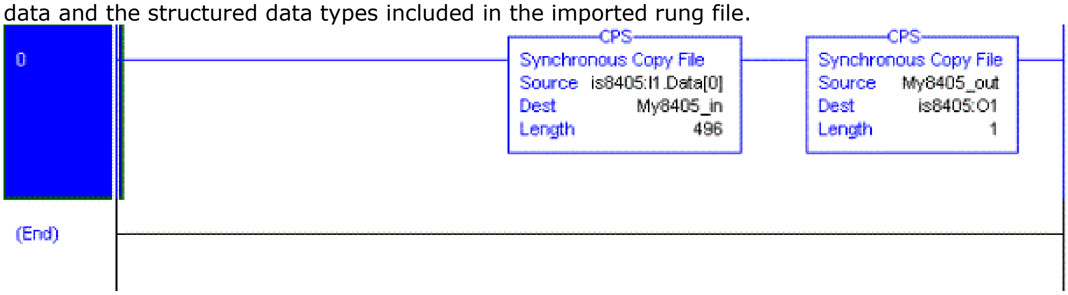


Import Name	Operation	Final Name	Alias For	Data Type	Description
insight11_in	Create	My8405_in		INSIGHT_I...	
insight11_out	Create	My8405_out		INSIGHT_...	
InSight_11:I1	Use Existing	is8405:I1		_02A6:is8xx...	
InSight_11:O1	Undefined	is8405:O1			

Name	Data Type	Description
is5705c:I1	_02A6:is57xx...	
is5705c:O1	_02A6:is57xx...	
is8405:I1	_02A6:is8xxx...	
is8405:O1	_02A6:is8xxx...	

Tag 'InSight_11:O1' is not defined in import file or in project.

8. Click the **OK** button of the Import Configuration dialog to close it. A rung will be added to the project, which includes the copy instructions for copying the data between the generic connection data and the structured data types included in the imported rung file.



Synchronous Copy File
Source is8405:I1.Data[0]
Dest My8405_in
Length 496

Synchronous Copy File
Source My8405_out
Dest is8405:O1
Length 1

When writing your ladder program, reference the elements in the structured data variables; the copy rung above will automatically move the data to and from the structured data instances to the EtherNet/IP connection. Download and run to get the tags to copy data initially, otherwise an array of SINTS will be all that is seen.

Establish the Connection Using a Generic ETHERNET-MODULE

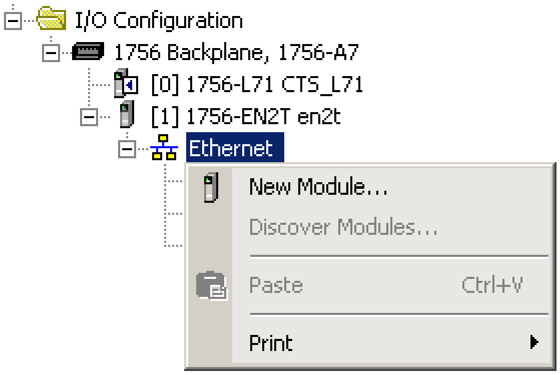
This topic covers integrating In-Sight vision systems running In-Sight 4.10.x or 5.x.x firmware in RSLogix 5000, version 14-19. These versions of RSLogix 5000 do not support the EDS files, and a Generic ETHERNET-MODULE must be used instead. In-Sight Explorer includes .L5X files, which can be used to copy PLC controller tag data to/from user-friendly program tags, representing signals in the In-Sight vision system running In-Sight 4.10.x or 5.x.x firmware.

> **Note**: The screen captures used in this topic were taken from RSLogix 5000, version 16. Some of the other versions of RSLogix 5000 may look slightly different.

1. Open RSLogix 5000 and load the PLC's project.

   > **Note**: The PLC must be Offline to add connections in RSLogix 5000.

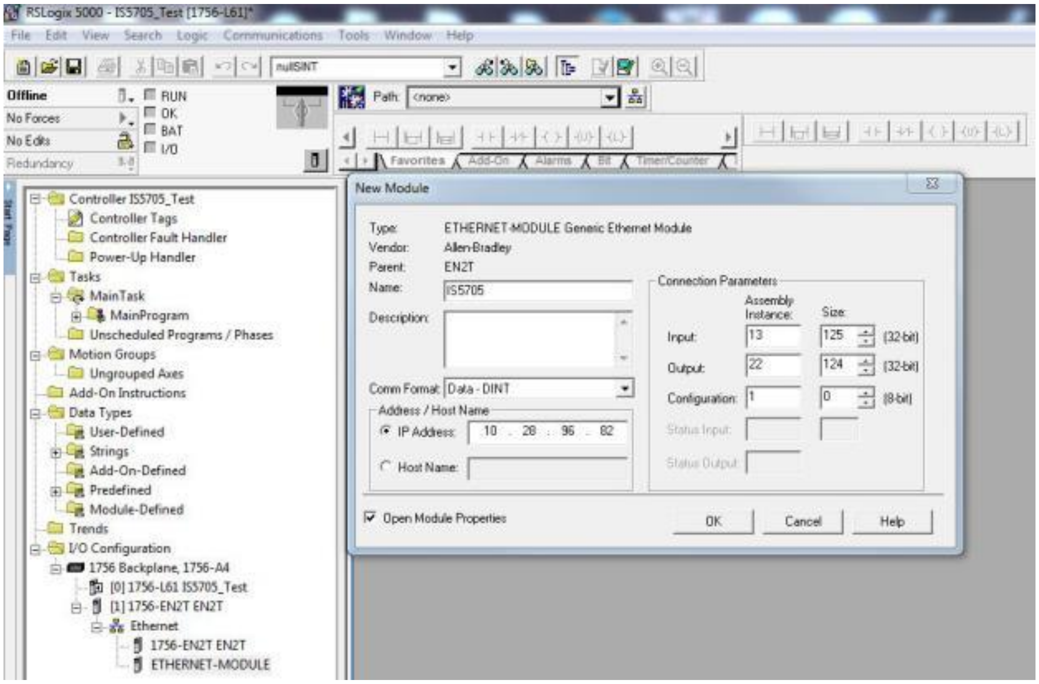
2. Add a connection to your Ethernet Communications Card. Under the I/O Configuration node, select the **Ethernet Node** under the Ethernet Module, right-click on the icon and select **New Module** from the menu.



3. From the *Select Module Type* dialog in the Catalog tab, select the **Communication** *Module Type Category Filter*, **Allen-Bradley** as the *Module Type Vendor Filter*, select **ETHERNET-MODULE** (Generic Ethernet Module) and press the **Create** button.

4. In the *New Module* dialog, enter the following:

   - The In-Sight vision system's name and IP address.
   - The Connection Parameters:
     - For In-Sight vision systems running In-Sight 5.x.x firmware, the Assembly Objects section in ISE HELP defines the *Assembly Instance* that should be configured.
     - For In-Sight vision systems running In-Sight 4.x.x firmware, the Assembly Objects section in ISE HELP defines the *Assembly Instance* that should be configured.

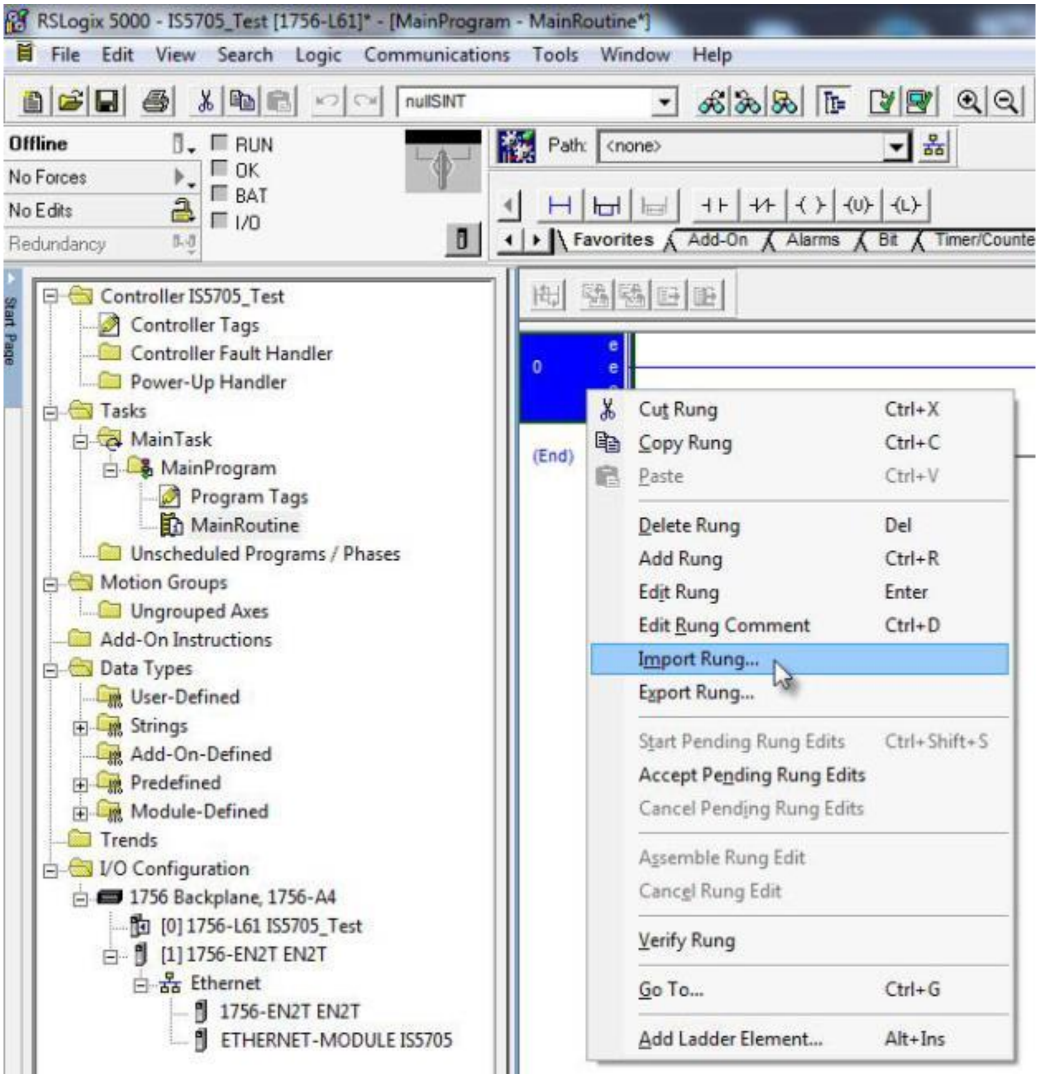


Type: ETHERNET-MODULE Generic Ethernet Module
Vendor: Allen-Bradley
Parent: EN2T
Name: IS5705
Description:
Comm Format: Data - DINT
Address / Host Name
● IP Address: 10 . 28 . 96 . 82
○ Host Name:

	Assembly Instance:	Size:	
Input:	13	125	(32-bit)
Output:	22	124	(32-bit)
Configuration:	1	0	(8-bit)

☑ Open Module Properties

5. After adding the vision system to the RSLogix project, the L5X rung import file can be used to create structured data for the connection.
    A. Within the main program, right-click on an empty rung and select the **Import Rungs...** option.

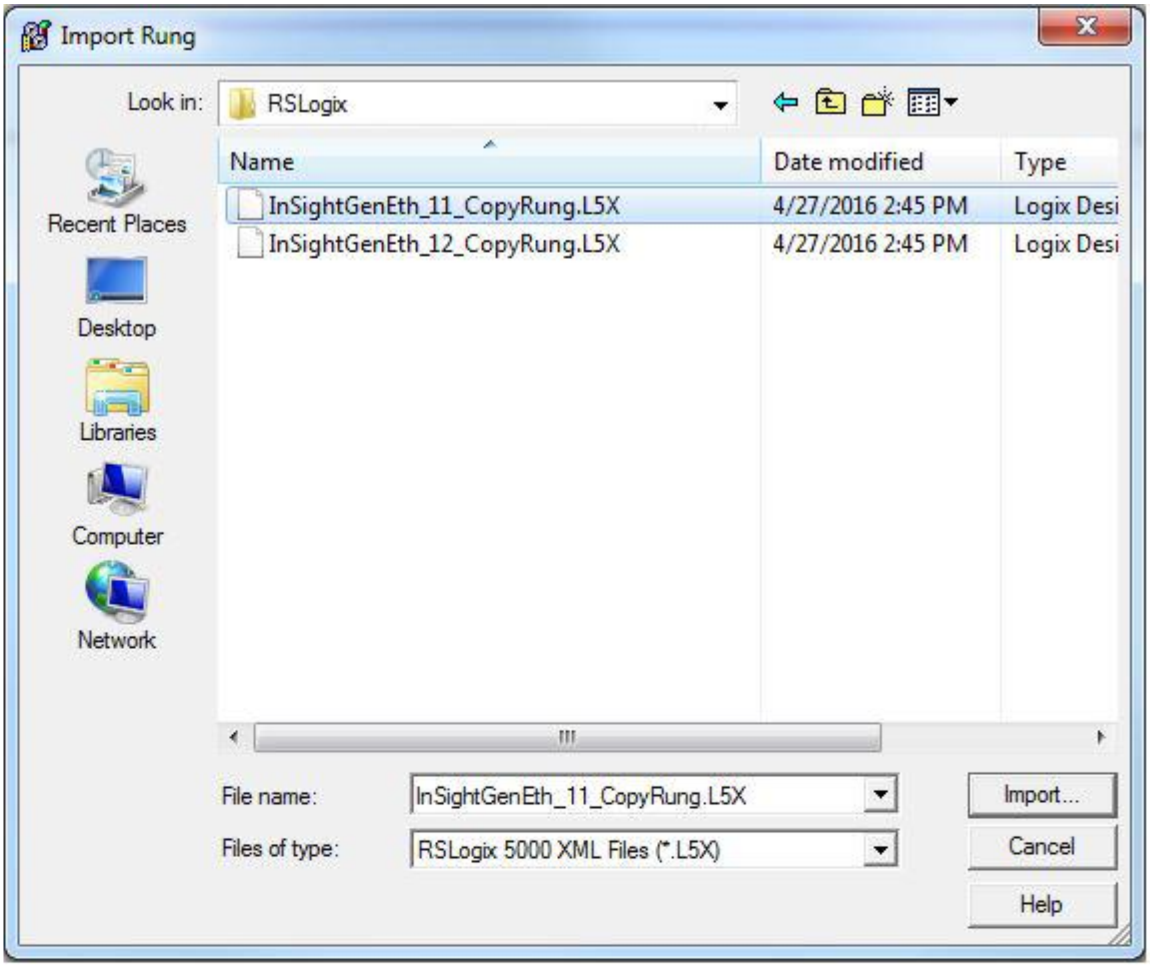


B. Navigate to the .L5X file and click the **Import...** button. If unsure of which file to use – review the "Integration with RSLogix" chart located at the beginning of this document.

- If using In-Sight 4.10.x firmware, and RSLogix version 14-16, import the **InSightGenEth_12_CopyRung.L5X** file.
- If using In-Sight 4.10.x firmware, and RSLogix version 17-19, import the **InSight_12_CopyRung.L5X** file.
- If using In-Sight 5.x.x firmware, and RSLogix version 14-16, import the **InSightGenEth_11_CopyRung.L5X** file.
- If using In-Sight 5.x.x firmware, and RSLogix version 17-19, import the **InSight_11_CopyRung.L5X** file.

InSight_12_CopyRung.L5X and InSight_11_CopyRung.L5X are included with the In-Sight Explorer install: C:\Program Files (x86)\Cognex\In-Sight\In-Sight Explorer 5.x.x\Factory Protocol Description\RSLogix.

InSightGenEth_11_CopyRung.L5X and InSightGenEth_12_CopyRung.L5X are included with this document. These files will be added to the In-Sight Explorer install with the ISE 5.4 release.



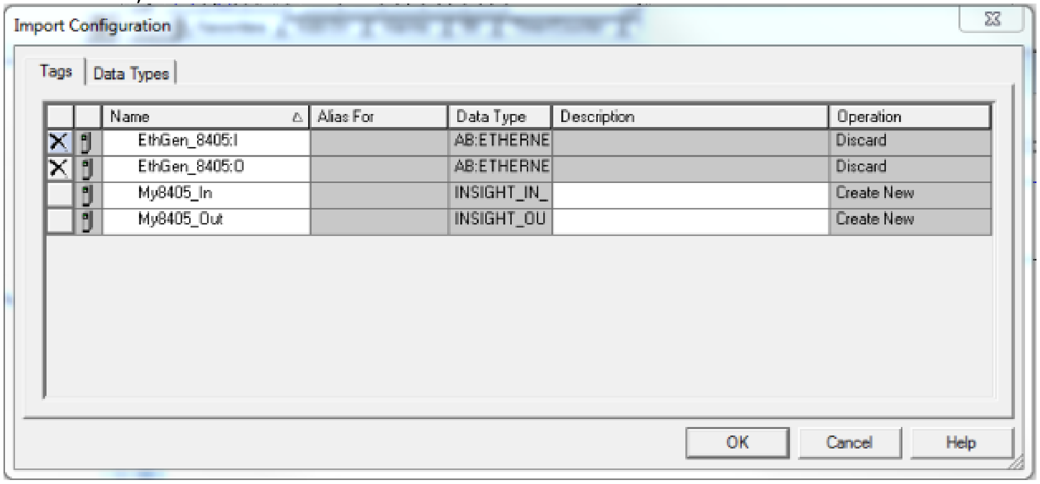
Look in: RSLogix

Name	Date modified	Type
InSightGenEth_11_CopyRung.L5X	4/27/2016 2:45 PM	Logix Desi
InSightGenEth_12_CopyRung.L5X	4/27/2016 2:45 PM	Logix Desi

File name: InSightGenEth_11_CopyRung.L5X

Files of type: RSLogix 5000 XML Files (*.L5X)

6. The *Import Configuration* dialog will then be used to change the names of the created tags to match the configuration.
    A. The generic names of the *Tags* will need to be modified. The top two lines will have an **X** beside them, and the *Name* of each will need to be modified.

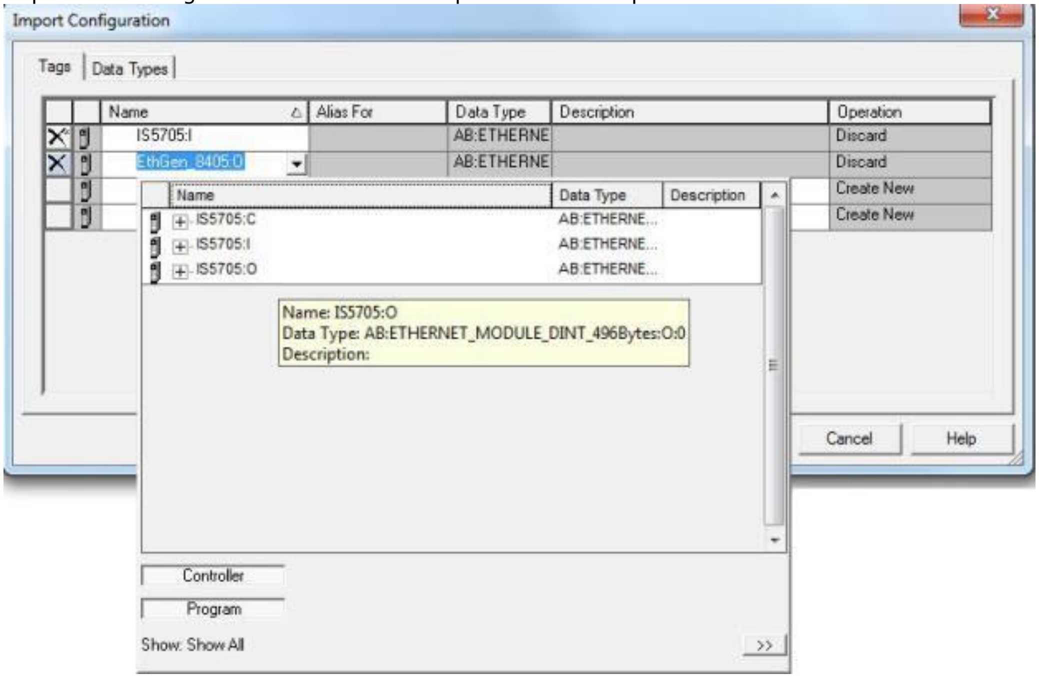


		Name	Alias For	Data Type	Description	Operation
X		EthGen_8405:I		AB:ETHERNE		Discard
X		EthGen_8405:O		AB:ETHERNE		Discard
		My8405_In		INSIGHT_IN_		Create New
		My8405_Out		INSIGHT_OU		Create New

    B. By clicking in the white space of the *Name* field, an arrow will appear, which will provide a drop-down menu that will contain replacement options for the default *Name* structure.

The list should contain names that match the name given in Step 4 (the name of the In-Sight vision system; in this example, the name given was IS5705, so the list contains IS5705:I and IS5705:O).

Replace the two generic names with the options in the drop-down menu.

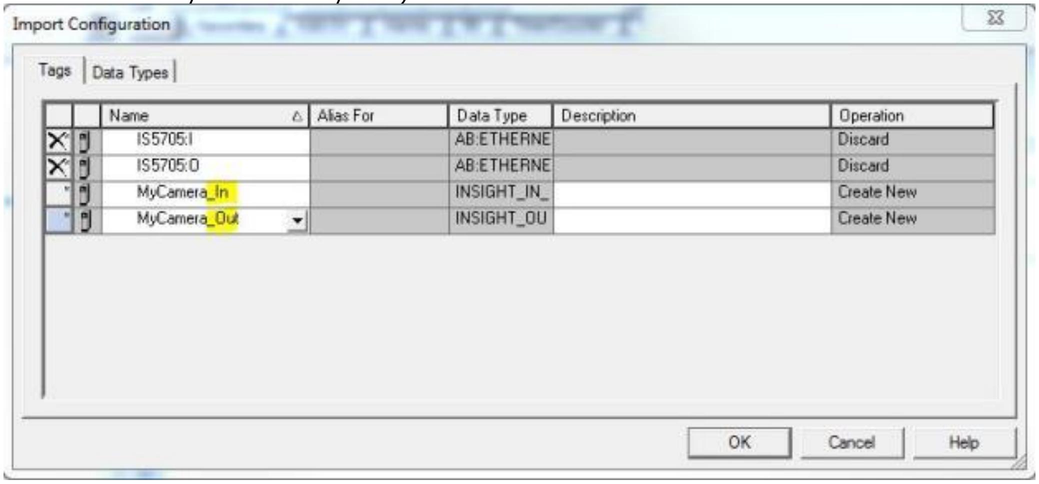


Name	Alias For	Data Type	Description	Operation
IS5705:I		AB:ETHERNE		Discard
EthGen_8405:O		AB:ETHERNE		Discard
				Create New
				Create New

Name	Data Type	Description
IS5705:C	AB:ETHERNE...	
IS5705:I	AB:ETHERNE...	
IS5705:O	AB:ETHERNE...	

Name: IS5705:O
Data Type: AB:ETHERNET_MODULE_DINT_496Bytes:O:0
Description:

C. The bottom two lines will also need to be edited. Click on the text of the name so that the cursor icon appears and then replace the text prior to "_In" and _Out" with the name of the generic module or some other easily identifiable name (e.g. "MyCamera" in this example, since there is only one vision system).

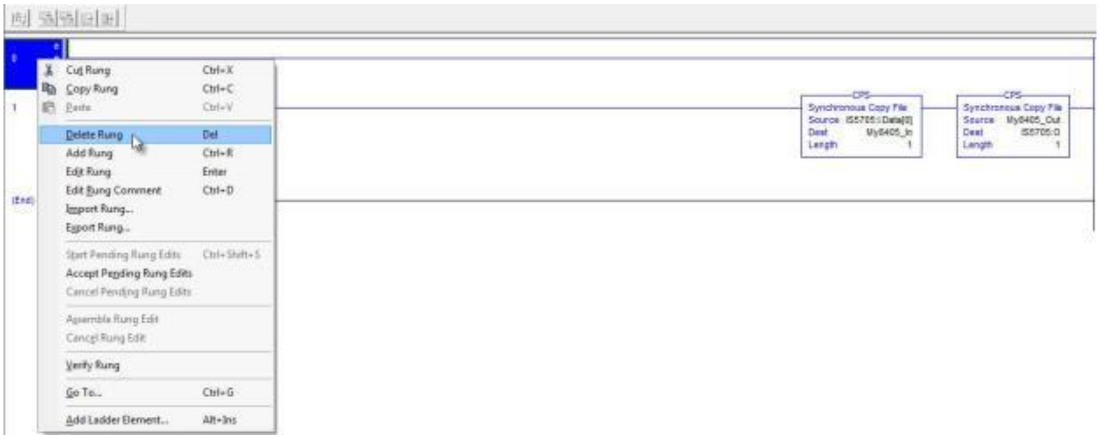


Name	Alias For	Data Type	Description	Operation
IS5705:I		AB:ETHERNE		Discard
IS5705:O		AB:ETHERNE		Discard
MyCamera_In		INSIGHT_IN_		Create New
MyCamera_Out		INSIGHT_OU		Create New

> **Note**: The name given in this instance should be unique to each vision system that is added to the I/O configuration. If multiple vision systems are going to be deployed, these steps must be followed for each vision system, because each instance of the copy rungs that are created will point to a specific vision system. Therefore, for later troubleshooting, ensure that the names used are easily identifiable.

D. Once all of the *Tags* have been updated from their default values, press the **OK** button.

7. A new rung will have been added to the *Main Routine* below the rung that was added (the empty rung can be deleted).



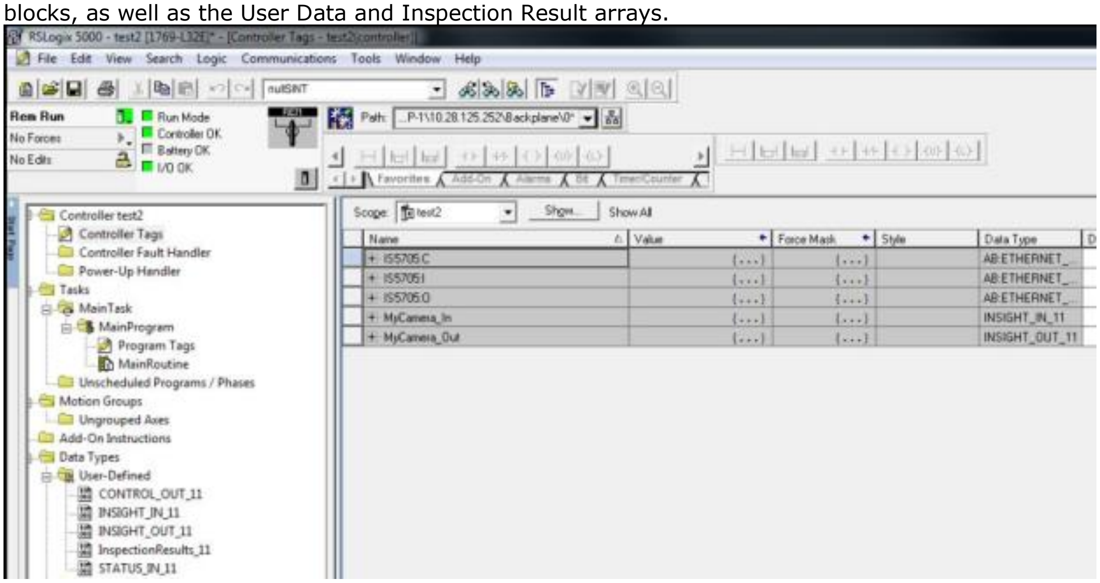
CPS
Synchronous Copy File
Source IS5705:I.Data[0]
Dest My8405_In
Length 1

CPS
Synchronous Copy File
Source My8405_Out
Dest IS5705:O
Length 1

8. The project can now be downloaded to the PLC, and the PLC can be put into **Run Mode**.

9. Once the rungs holding the copy instructions have been run, the new tags will be created in the *Controller Tags* table.

   In this example, the *IS5705:I* and *IS5705:O* tags are the raw input and output assemblies from the vision system, and the *MyCamera_In* and *MyCamera_Out* will contain the Control and Status blocks, as well as the User Data and Inspection Result arrays.



Name	Value	Force Mask	Style	Data Type
+ IS5705:C	{...}	{...}		AB:ETHERNET_...
+ IS5705:I	{...}	{...}		AB:ETHERNET_...
+ IS5705:O	{...}	{...}		AB:ETHERNET_...
+ MyCamera_In	{...}	{...}		INSIGHT_IN_11
+ MyCamera_Out	{...}	{...}		INSIGHT_OUT_11